LA PRIMA RIVISTA EUROPEA DI SOFTWARE PER PERSONAL COMPUTER

PERSONAL SOFTWARE

In omaggio la "Guida al software" pubblicato sulle riviste italiane

ANNO 1 N. 2 SETTEMBRE 1982 L. 3.500

UNA PUBBLICAZIONE DEL GRUPPO EDITORIALE JACKSON

- PROGRAMMI COMPLETI PER APPLE, PET/CBM TRS/80, SINCLAIR E VIC 20
- IMMAGINI DIGITALI DI MEZZE TINTE
- RICERCA SU ALBERI
- UN ASSEMBLATORE IN BASIC PER IL PET/CBM
- L'ORGANIZZAZIONE PERT

Spedizione in abb. postale Gruppo III/70

A Warner Communications Company

COSA FA UN COMPUTER ATARI CON TUO FIGLIO?

Quali stati compongono gli U.S.A.? Come si dice 'Ci vediamo lunedì' in francese? Atari studia con tuo figlio nel modo più moderno e intelligente. Con Atari lui impara a risolvere problemi di ogni tipo: da come sarebbe andata a Waterloo se lui avesse diretto le operazioni, a come riprodurre un brano musicale, a come risolvere una equazione. E si prepara così alla nuova civiltà del computer.

Non per niente Atari è in dotazione in molte scuole di ogni tipo, come aiuto prezioso sia per gli studenti che per gli insegnanti.

Non avevate ancora pensato di regalare un computer a vostro figlio? Allora, pensate a un Atari. E' facile e veloce, non più grande di una macchina per scrivere e non più costoso di un hi-fi. Basta collegarlo a un qualsiasi apparecchio TV e Atari è pronto per funzionare. A casa vostra, magari anche per giocare con i videogames, ma naturalmente anche nel vostro studio o negozio, officina o laboratorio.

Perchè Atari sa fare molte cose, è il nuovo strumento di lavoro per chi vive la professione in modo moderno e aggiornato. Atari calcola, prevede, consiglia soluzioni, disegna grafici e figure, archivia, fattura... è il collaboratore-amico che vi può dare una mano ogni giorno.

Potete sceglierlo nel più agile modello 400 o, se avete problemi più complessi, nel sofisticato modello 800, fornito del potente sistema gestionale VisiCalc. E potete scegliere anche tra una vastissima gamma di accessori.

Chiedete un Atari in prova al vostro negoziante di fiducia, e dopo un giorno insieme vedrete come è utile, facile e, perchè no, affascinante possedere un Atari.

ATARI®
Computers for people.

DISTRIBUTORE ESCLUSIVO PER L'ITALIA
ADVEICO
CONSUMER DIVISION

ADVEICO S.p.A. - 20124 Milano - Via A. Tadino, 22, Tel. 02/2043281 - 43016 S. Pancrazio, Parma - Via Emilia Ovest, 129, Tel. 0521/998841

ELEDRA PERSONAL COMPUTER NEWS

GIUGNO 1982 1

PUBBLICAZIONE GRATUITA DEL GRUPPO ELEDRA

1982: Terza generazione del Personal Computer. L'era della computerizzazione di massa è appena iniziata... non solo per l'utente ma anche per gli operatori come noi. La strada da percorrere è lunga e i problemi da risolvere sono molti, per questo riteniamo urgente e fondamentale impostare con tutti gli interessati un **corretto colloquio.**

Il **gruppo Eledra,** nato nel '66 con il preciso scopo di commercializzare e assistere tecnologie nuove, è stato il primo, dieci anni or sono, a introdurre in Italia il capostipite dei personal computer: il microcomputer Intel 4004.

Abbiamo istituito corsi e seminari, prodotto testi specializzati e fornito tutti gli indispensabili servizi oggi utilizzati dall'industria elettronica italiana.

Nello stesso modo vogliamo fornire servizi di effettiva utilità al potenziale utente di personal computer impostando contemporaneamente un **dialogo attraverso l'Eledra Personal Computer News.**

Se siete interessati potete farne richiesta utilizzando la cartolina allegata o scrivendoci direttamente.

Riceverete così gratuitamente informazioni sui prodotti da noi scelti, notizie su questo mercato e sui servizi disponibili.

La nostra organizzazione, con i suoi 5 centri regionali, oltre 120 persone e numerosi punti di vendita autorizzati è a Vostra disposizione anche solo per darVi informazioni.

ELEDRA 3S spa · Viale Elvezia, 18 · 20154 Milano

il futuro nero su bianco

RICHIESTA DI ABBONAMENTO GRATUITO

Spedire il coupon in busta chiusa a:

ELEDRA 3S S.p.A , - Viale Elvezia, 18 - 20154 Milano

☐ Desidero ricevere regolarmente Eledra Personal Computer News

☐ Desidero essere visitato da un vostro funzionario

P52

Cognome e nome ____________________

____

Attività ____________________

Indirizzo ____________________

Città ____________________ Tel ____________________

cambiate disco!

FLEXETTE
N°
Code 4001
RHÔNE-POULENC

SMAU 82
Pad. 14 - Sal. 3
Stand: R17
VI ASPETTIAMO

RHÔNE-POULENC SYSTEMES
FLEXETTE

Rispondere alle crescenti esigenze degli utilizzatori di mini e macro sistemi è la missione che si è fissata RHONE-POULENC SYSTEMES fabbricando FLEXETTE: la nuova famiglia di Floppy discs, disponibili in 8" e 5" 1/4 fabbricati da DYPY S.A. (*).
FLEXETTE è oggetto di controlli permanenti e di testi unitari che lo garantiscono REALMENTE ERROR-FREE.

Dalla sua prima utilizzazione Voi siete sicuri di ottenere registrazioni ad alta fedeltà. D'altra parte la finitura della superficie del disco, la certificazione al 100% di questa superficie permettono di ottenere delle condizioni di registrazione eccezionali.
FLEXETTE è riservato agli utilizzatori che ricercano la garanzia di un'alta tecnologia.

Cambiate disco, provate FLEXETTE e approfittate di tutte le sue qualità.

RHÔNE POULENC ITALIA S.p.A.
Divisione Rhône Poulenc systemes
Via Romagnoli, 6 - 20146 MILANO tel. 42461
telex ITARPC 332330

PERSONAL SOFTWARE

ANNO 1 N. 2 SETTEMBRE 1982

Sommario

RUBRICHE

ARTICOLI

PROGRAMMI

TechnoClub

PRESENTA

applemail*

Finalmente risolto il problema delle circolari e delle lettere personalizzate.

Con Apple Writer puoi comporre il testo della circolare

Con il Personal Data Base generare archivi o utilizzare quelli che già usi per altri scopi

Con estrema semplicità AppleMail consente di creare circolari o lettere personalizzate

L'estrema semplicità di utilizzo rendono questo package lo strumento ideale per redigere circolari promozionali, lettere personalizzate, estratti conto periodici o qualunque altro documento contenente dati e indirizzi.
Tutto ciò senza dover creare archivi complementari, ma sfruttando quelli già creati e utilizzati normalmente per altri servizi. Il programma, fornito completo di manuale d'uso, floppy disk ed elegante contenitore, può essere acquistato al prezzo di L. 150.000 (IVA compresa) presso uno dei rivenditori Apple oppure presso TechnoClub inviando il coupon riprodotto in questa pagina.
In entrambi i casi diventerai automaticamente socio del TechnoClub con tutti i vantaggi illustrati nelle pagine precedenti.
Possiamo inoltre fornire i programmi:
Apple Writer a L. 75.000
Apple Personal Data Base a L. 60.000

Cognome ....

Nome ....

Via .... N ....

Città .... C.A.P. ....

C.F. (per le aziende) ....

Confermo l'acquisto di

N .... package AppleMail a L. 150.000 cad.
N .... package Apple Writer a L. 75.000 cad.
N .... package Personal Data Base a L. 60.000 cad.

DATA .... FIRMA ....

scegliendo la seguente forma di pagamento:

- ☐ Allego assegno N. .... di L. ....
- ☐ Ho versato l'importo tramite vaglia postale di cui allego ricevuta
- ☐ Ho versato l'importo sul CCP n. 19445204 intestato a TechnoClub - Milano

NON SI EFFETTUANO SPEDIZIONI IN CONTRASSEGNO.

*Apple è un marchio registrato della Apple Computer Inc. Cupertino - USA

TechnoClub - Via Rosellini, 12
20124 Milano - Tel. (02) 6888228

Le mani sulla tastiera

Mauro Boscarol

Qualcuno mi ha fatto sapere di non essere d'accordo con quello che ho scritto nell'editoriale del primo numero, ed ha espresso il suo punto di vista dicendo che "anche i programmi sono software".

Non ho mai pensato il contrario. I programmi lo sono certamente, ma quello che avevo scritto e che ora ripeto è che il software non sono solo i programmi.

Questo lo sa molto bene chiunque abbia in qualche modo messo le mani sulla tastiera di un computer provando a realizzare qualcosa. Costui conosce le riflessioni e lo studio che precedono la stesura dell'algoritmo, le molte modifiche, la scrittura del programma, le esecuzioni di prova, il *debugging*. Ecco, per esempio, il *debugging* è senz'altro un aspetto peculiare del software, ma non si può dire che si tratti di "programmi". È una cosa diversa, un insieme di tecniche di correzione e messa a punto, fondamentale in informatica. Parlando in termini matematici, ma ormai accessibili anche agli scolari delle elementari, i programmi sono un sottoinsieme (proprio) del software.

Oltre ad essi ci sono varie tecniche e settori di interesse che costituiscono materia di studio e riflessione: per esempio l'analisi degli algoritmi, la loro complessità, la loro correttezza.

In questo numero c'è un articolo che ben si presta ad esemplificare. Si tratta di *Immagini digitali di mezzetinte*. Qui sono presentati dei programmi, di per sé molto semplici e, se si vuole, banali. Ma il loro significato, la loro applicazione al problema di riproduzione dei toni sfumati di bianco e nero, gli studi che ci stanno sotto, costituiscono un problema di software generale, non un problema di programmazione.

Si verifica in questo campo ciò che ho personalmente sperimentato nel mondo universitario. Per i matematici astratti (quelli, per intenderci, che si interessano di analisi, algebra, geometria) tutti i matematici applicati che lavorano con il computer fanno "analisi numerica". Non importa se si interessano di informatica teorica, di basi di dati o grafica con il computer. Probabilmente ciò si può spiegare con il senso di sicurezza che danno le classificazioni: se dico che il software sono i programmi, l'ho incasellato e lo posso dominare.

Coloro che non si accontentano di questo, vogliono vedere oltre. E la nostra rivista dà loro una mano con gli articoli, le rubriche, e (anche) i programmi.

Cosa c'è di nuovo questo mese? Fra gli altri articoli vorrei segnalare la prima parte di *Ricerca su alberi* che tratta in dettaglio questa particolare tecnica di tipo combinatorio molto utilizzata in intelligenza artificiale. Quindi *L'organizzazione PERT* un breve ma chiarissimo articolo su questa tecnica fondamentale della ricerca operativa ed infine un editor e assemblatore in Basic per il PET/CBM serie 3000 e 4000.

Vorrei concludere con un invito. Le pagine di questa rivista sono aperte a tutti. Se avete un articolo che pensate sia nello spirito e nella filosofia di queste nostre pagine, fatecelo avere. Se pensate di avere qualcosa da dire in merito agli argomenti trattati nelle rubriche o in questo stesso *Editoriale*, scrivete voi stessi un numero della rubrica che vi interessa. Se pensate di aver scritto un programma interessante e utile, mandatecelo. La vostra collaborazione sarà ricompensata, ma soprattutto potremo aprire un dialogo tra tutti noi. Cosa che comunque inizieremo a fare dal prossimo numero con la rubrica della posta. Scriveteci.

Nel prossimo numero:

Ricerca su alberi, seconda parte

Compressione di testi

Generatore di labirinti

Comunicazioni da PET a PET

SHARP MZ-80B + CP/M*
Con questa semplice addizione

il Personal più versatile diventa ora il più completo.

Lo **MZ-80B** è il personal realizzato dalla Sharp per gli ingegneri edili, per i responsabili di produzione, per i progettisti, per i chimici, per i ricercatori, allo scopo di aiutarli nel loro lavoro di progettazione, di calcolo, di controllo, di ricerca e di analisi. La sua versatilità di impiego è ormai proverbiale: lo **MZ-80B** è infatti dotato di capacità grafiche di prim'ordine che permettono di visualizzare sul suo monitor ad alta definizione (200 x 300 punti) situazioni statiche o dinamiche.

Lo **MZ-80B** non è inoltre legato ad un linguaggio residente su ROM, ma è possibile caricare di volta in volta in RAM diverse versioni di BASIC, interpreti e compilatori, del Pascal, dell'ASSEMBLER, eccetera.

SHARP

Per rendere lo **MZ-80B** ancora più completo, la Sharp ve lo offre ora dotato del sistema operativo CP/M. Lo **MZ-80B** ha così accesso a una ricchissima biblioteca di linguaggi e di programmi applicativi, per il word processing, per i modelli finanziari, per la gestione del data base, per citarne solo alcuni. Questa biblioteca va ad arricchire la già ricca dotazione di pacchetti applicativi realizzati dalla Melchioni Computertime.

Il CP/M non si limita a rendere più completa la dotazione di software dell'**MZ-80B** ma lo rende anche più potente perchè porta a 680 kbyte la capacità del doppio drive di dischetti del sistema.

**CP/M è un nome depositato della Digital Research Ltd.*

MELCHIONI COMPUTERTIME

Concessionari e Rivenditori autorizzati presenti in ogni provincia italiana

Via Fontana, 22 - Milano - Tel. 585.116-541.569

SHARP COMPUTERS.

I Nobel dell'informatica.

Ricerca su alberi

Prima parte: tecniche di base

di *Gregg Williams*

È stato stimato che il numero di strategie nel gioco degli scacchi supera quello degli atomi dell'universo. Ciò significa che se un computer potesse generare un milione di mosse al secondo, impiegherebbe circa 3.2×10^{60} *secoli* per produrre tutte le strategie possibili. Come sono in grado, allora, i giochi di scacchi basati su microprocessori da 200 dollari (che devono affrontare l'analisi di una situazione così complessa) di elaborare anche strategie particolarmente valide? Tra le tecniche necessarie a questo scopo una delle più potenti, nel campo dell'intelligenza artificiale, è conosciuta come *ricerca su alberi*.

Essa consente ad un calcolatore di scegliere la migliore tra tante alternative, considerando però il minor numero possibile di soluzioni parziali. La prima parte di questo articolo affronta le tecniche di base della ricerca su alberi a tre stadi:

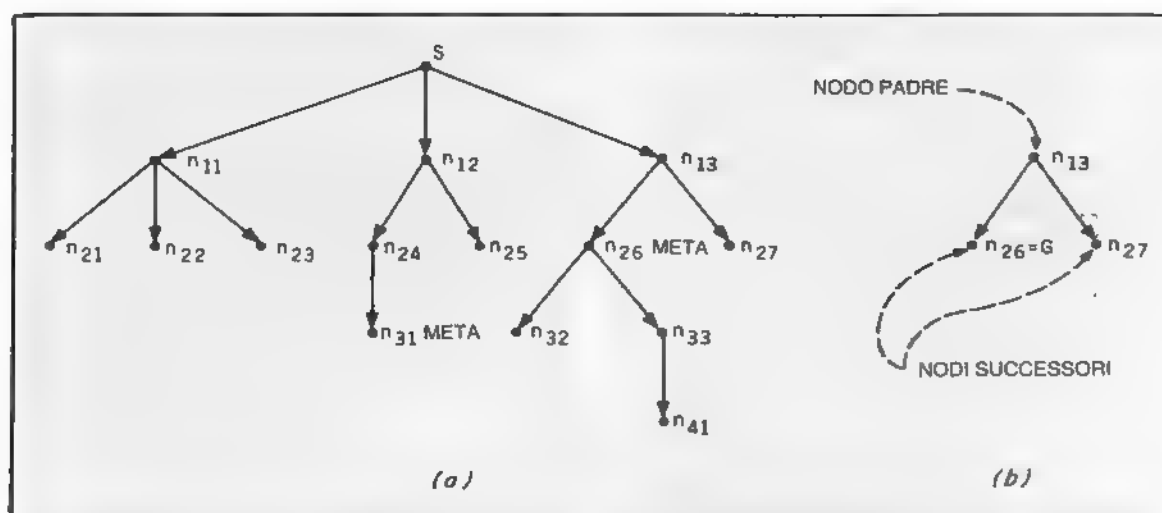



Fig. 1. *Nomenclatura degli alberi. La figura 1a è la rappresentazione grafica ad albero dello spazio degli stati di un problema. Gli alberi sono caratterizzati dall'avere un unico nodo di partenza (S), dal contenere solo nodi derivati da quello iniziale e dall'assenza di freccie dirette a nodi dello stesso livello o di livelli inferiori. I nodi n_{21}, n_{32} ed n_{27} sono tre esempi di nodi terminali; i nodi n_{12}, n_{26} ed n_{33} sono invece esempi di nodi intermedi; i nodi n_{31} ed n_{26} sono mete. Si noti che una meta può essere sia nodo terminale che intermedio. La figura 1b mostra la relazione tra il nodo padre n_{13} ed i successori n_{26} ed n_{27}. Si dice che il nodo n_{13} si* estende *a generare n_{26} ed n_{27}.*

La riproduzione di questo articolo è stata concessa da BYTE. *Traduzione di Flavio Santini*

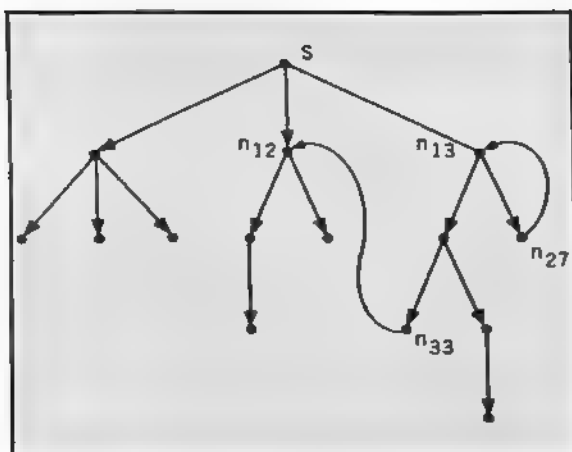



Fig. 2. *Un grafo orientato. Questa figura, che rappresenta la struttura di molti problemi "reali", differisce da quella di un albero, poiché qui i successori possono essere allo stesso livello o ad un livello "più basso". Il successore di* n_{33} *è* n_{12}*; il successore di* n_{27} *è* n_{13}*. Gli algoritmi di ricerca su alberi possono essere modificati per trattare, in generale, i grafi, ma questo articolo si occupa solo degli alberi.*

teoria, implementazione (attraverso diversi programmi in Basic che illustreranno le tecniche principali) e sperimentazione. Sarà introdotta la terminologia fondamentale e mostreremo alcuni ben noti metodi *esaustivi* di ricerca su alberi (quelli in grado di generare tutte le possibili soluzioni parziali), assieme ad un programma Basic illustrativo (per risolvere il noto gioco dei 15 numeri da ordinare in 16 caselle) che verrà usato anche quando si tratteranno i *metodi euristici possibili* di ricerca che si servono di informazioni sul sistema esaminato per ridurre il numero di percorsi sbagliati da esaminare; ci si occuperà anche di tecniche euristiche non ammissibili che cercano una soluzione più rapida a costo di perdere la certezza dell'*ottimalità* della soluzione, o rischiando addirittura di non trovarne alcuna.

Terminologia di base

Il fine dell'intelligenza artificiale, secondo una scuola di pensiero, è di produrre programmi per computer che risolvano problemi non facilmente trattabili da una macchina – problemi che si prestano ad essere risolti da una componente "intelligente" (di solito un essere umano). La soluzione a molti di questi problemi può essere vista come la ricerca di una *meta*, che ha proprietà definite in modo non ambiguo, a partire da uno stato (o *nodo*) iniziale e seguendo un certo insieme di regole. Tra il nodo iniziale e la meta ci sono altri nodi, che rappresentano le posizioni intermedie. Noi siamo interessati a quella particolare sequenza di nodi che costituisce il percorso minimo verso una delle possibili mete. Molti nodi (generalmente un numero enorme) non giacciono sul cammino ottimale, e lo scopo dei metodi di ricerca sull'albero è quello di esplorare il *minor* numero possibile di essi.

Parecchi problemi che non sembrano prestarsi alla ricerca su alberi si riconducono a questo genere quando sono descritti secondo una *rappresentazione a numero finito di stati*. Talvolta, nel caso di problemi che hanno campo d'esistenza continuo (e quindi infinito), ciò si traduce nel ridurre il problema ad un numero finito di passi discreti: un esempio si ha considerando l'inter-

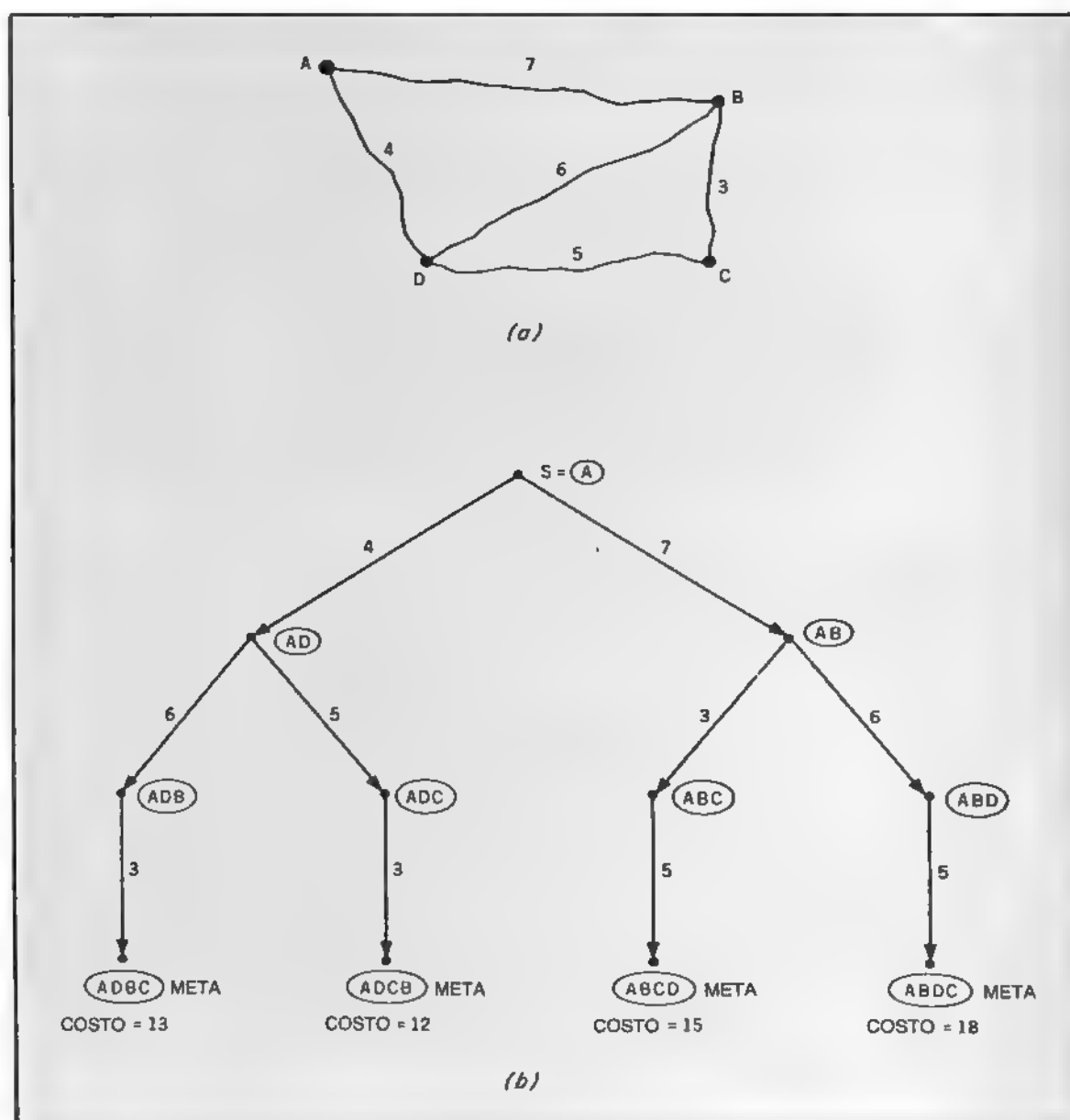



Fig. 3. *Il problema del commesso viaggiatore. Data la mappa di figura 3a, lo scopo è di trovare il percorso minimo che, partendo dalla città A, passi per le altre B, C e D. L'albero della figura 3b dà la rappresentazione a numero finito di stati del problema, dove ogni nodo è una tappa del percorso completo (cioè: ADC è il cammino da A a C attraverso D) il cui costo dipende dalla strada seguita. In questo caso il problema è discreto ed ogni nodo ha un numero finito di successori; dal nodo AD sono possibili solo due tragitti: ADB ed ADC.*

SORD M23 LAVORA IN PIPS NON-STOP

PIPS

Il nuovo non-linguaggio di programmazione che ha reso il computer accessibile a tutti.

NON-STOP

M23 il microcomputer facile ed affidabile con una grande flessibilità di impiego, che trova limitazioni solo nella propria fantasia. Ha la capacità di lavorare senza interruzioni, a lungo. Se si ferma è per fatti eccezionali. L'ultima volta c'era un topolino dentro!

SORD M23

128K Ram - Video 12" 14" verde-arancio-colore 2 floppy 5" 1/4 per 660Kbytes
2 porte seriali - 1 porta parallela - Basic - interprete - compilatore Pascal, Fortran, Cobol
Standard il nuovo modo di programmare Pips

Lit 4.900 000 + I.V.A. Prezzo 'tutto compreso

Garanzia per un anno e speciale polizza assicurativa

Si cercano rappresentanti per zone libere

cattaneo system
Via Cesarea, 9/4 16121 Genova (Italy
Tel (010) 595852/51 Telex 271225

Importatore esclusivo

SORD

Sord computer systems, inc.

Per maggiori informazioni inviare il tagliando a
cattaneo system spa via cesarea 9/4 - 16121 genova

nome
indirizzo
cap città
tel.
professione

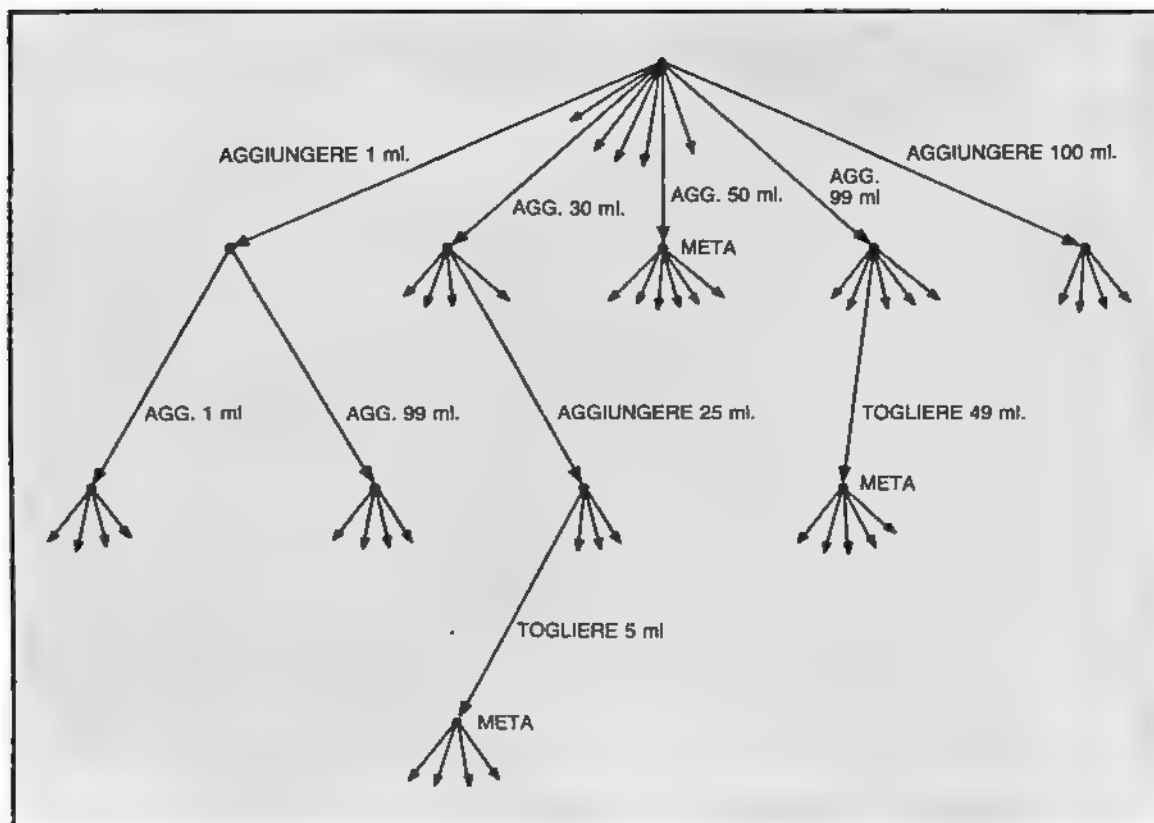



Fig. 4. *La "quantificazione" di uno spazio degli stati continuo. È illustrato un pezzo dell'albero corrispondente al problema di riempire un bicchiere fino ad un livello stabilito. Lo spazio (la quantità d'acqua nel bicchiere) è continuo nel senso che il problema può aver a che fare con un'infinita varietà di stati (quantità d'acqua). Se consideriamo di riempire il bicchiere solo con incrementi millilitro per millilitro, lo spazio del problema diventa discreto e finito, e così può essere trattato col calcolatore. Le mete si hanno quando l'acqua raggiunge il livello fissato.*

vallo delle temperature da 20 a 30 gradi centigradi come insieme discreto di valori – diciamo 20.0°, 20.1°, 20.2°,..., 29.9°, e 30.0°. In teoria questo comporta una perdita di precisione, ma molti problemi possono essere rappresentati secondo incrementi così piccoli, che la precisione è salvaguardata.

Una rappresentazione a numero finito di stati è formata da: un *nodo iniziale*, una meta o un *insieme di mete*, e delle regole o *operatori* che permettono, dato un certo nodo, di generare tutti i nodi successori possibili. In certi casi, al percorso del nodo ad un suo *successore* (nodo generato da una singola applicazione di un operatore al *nodo padre*) è associato un *costo*, ed il costo di una meta è il costo totale del percorso minimo dal nodo iniziale alla meta stessa. L'insieme di tutti i nodi che possono seguire quello iniziale è chiamato *spazio degli stati*. Un *albero* è una rappresentazione possibile dello spazio degli stati di un problema. Come si vede nella figura 1, i nodi vengono rappresentati con punti, e la relazione tra un certo nodo ed un suo successore è schematizzata da una freccia diretta dal padre al successore. Chiameremo S il nodo di partenza ed ogni altro nodo secondo il suo *livello* (distanza dal nodo iniziale) e la sua posizione all'interno dell'insieme dei nodi con lo stesso livello (si tratta di un etichettamento arbitrario, ma semplice e molto utile). Con riferimento alla figura 1a, i nodi n_{11}, n_{12} ed n_{13} sono tutti quelli di livello 1. Il nodo n_{12} ha due successori, n_{24} ed n_{25}. Il nodo n_{41} è l'unico con livello quattro.

L'applicazione degli operatori, che determinano la transizione da uno stato (nodo) al successivo, a certi nodi non dà luogo ad altri stati. Questi nodi sono detti *terminali*; i nodi terminali nella figura 1a sono n_{21}, n_{22}, n_{23}, n_{25}, n_{27}, n_{31}, n_{32} ed n_{41}. I nodi segnati con la parola META o la lettera G sono le mete; e possono essere sia nodi terminali che intermedi.

Spesso lo spazio degli stati di un problema consente ad un nodo di generarne come successore un altro di livello uguale o minore, come nella figura 2. In questo caso, la rappresentazione che ne risulta è chiamata *grafo orientato*. Questo richiederà solo una lieve modifica al metodo di ricerca su alberi, ma è importante notare la differenza.

Infine, come abbiamo già accennato, un certo costo può essere associato alla transizione da un nodo al successivo. In tal caso, il suo valore viene scritto accanto alla freccia corrispondente; altrimenti la freccia non è etichettata e si suppone che abbia costo unitario.

Alcuni esempi

Analizzeremo dapprima un esempio discreto: il problema del commesso viaggiatore. Un commesso viaggiatore abitante nella città A deve passare per le città B, C e D in un ordine qualsiasi. Data la mappa in figura 3a, quale percorso attraverso questi centri minimizza la lunghezza totale percorsa?

In questo caso, i nodi sono percorsi parziali (o completi) descritti da una sequenza di lettere A, B, C e D, con le seguenti regole. Il nodo iniziale è A (la sede è la città A, nessun viaggio). Quattro mete sono rappresentate dai percorsi completi ABCD, ABDC, ADBC, ADCB. Informalmente le regole possono essere riassunte così: ad un certo nodo, aggiungere la lettera corrispondente ad una qualsiasi città connessa con quella rappresentata da quel nodo, purché non ancora visitata.

Poiché il numero delle città è piccolo, possiamo tracciare l'intero albero rappresentante lo spazio degli stati (figura 3b), da cui risulta chiaramente che il percorso minimo completo è ADCB, con lunghezza (costo) dodici. Ma se le città fossero dieci? O venti? E se alcune strade fossero a senso unico?

Vediamo ora un esempio nel continuo (dove lo spazio degli stati dev'essere ridotto ad un numero finito di valori discreti). Supponiamo di avere un rubinetto, uno sca-

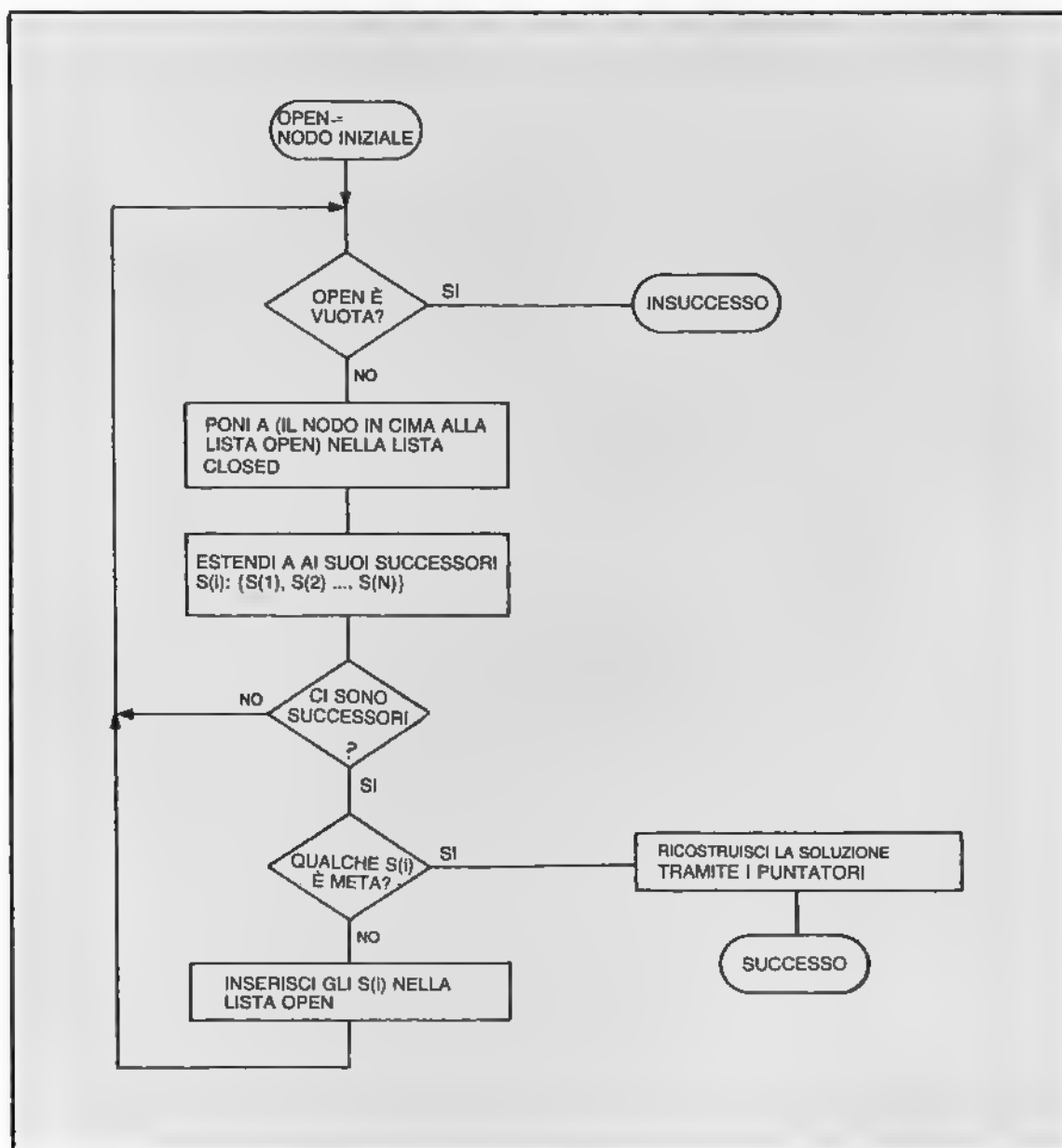



Fig. 5. *Il flowchart di base usato per scrivere il programma RICERCA del listato 1a. Prima di scrivere in Basic il programma RICERCA, il flowchart è stato tradotto in pseudolinguaggio (vedi listato 1b).*

rico ed un bicchiere vuoto con una linea orizzontale: bisogna riempire il bicchiere fino alla linea. Prima di tutto dobbiamo quantificare il problema. Supponiamo che il bicchiere contenga 100 ml di acqua. Possiamo anche considerare il millilitro come la più piccola quantità d'acqua trattabile. Gli stati del problema sono le varie quantità d'acqua presenti nel bicchiere, e possono essere 101: 0 ml (vuoto), 1 ml d'acqua, 2 ml,..., 100 ml. (Qui la parola "stato" è più appropriata di "nodo"; si usa il secondo termine quando ci si riferisce alla rappresentazione grafica del problema.)

Ma, oltre al volume, bisogna quantificare il tempo, nel senso che si aggiunge o si toglie una certa quantità d'acqua "alla volta". Gli operatori possibili sono 200 (alcuni dei quali fisicamente irrealizzabili a partire da un certo stato): aggiungere 1 ml, togliere 1 ml, aggiungere 2 ml,...e così via fino ad aggiungere o togliere 200 ml. Infine, la meta è un qualsiasi insieme di operazioni che porti il livello dell'acqua alla linea tracciata sul bicchiere (dovunque essa sia). L'albero corrispondente allo spazio degli stati è grande ma finito; parte di esso è mostrata nella figura 4.

Il gioco del 15

Molti di voi avranno già giocato al gioco del 15: quindici quadratini numerati che si muovono all'interno di uno schema quadrato con lato di quattro unità. I quadratini sono all'inizio disordinati e lo scopo è quello di spostarli finché, letti riga per riga, siano in ordine crescente con lo spazio vuoto nell'angolo in basso a destra. (Il matematico Sam Lloyd fece un mucchio di soldi scommettendo con altri che non avrebbero risolto il giochetto. Andava sul sicuro; partendo dalla situazione che lui proponeva, era impossibile trovare la soluzione.)

Useremo il gioco del 15 per illustrare vari metodi di ricerca. Il programma Basic di nome RICERCA (vedi listato numero 1) implementerà diverse tecniche di ricerca al variare di un'unica subroutine. È valido sia il gioco del 15 standard che (per i computer con poca memoria) per la versione con lato di 3 quadratini: il gioco dell'8. In questo articolo parleremo proprio del gioco dell'8; anche con uno schema 3 per 3, l'albero su cui effettuare la ricerca si ramifica velocemente.

Strategia elementare

Per implementare qualsiasi strategia di ricerca c'è bisogno di un certo formalismo. Useremo due liste chiamate OPEN e CLOSED, la prima per i nodi che non si sono ancora estesi (cioè che non hanno generato tutti i possibili successori) e la seconda per quelli già estesi; un algoritmo per generare tutti i successori consentiti; ed un algoritmo per determinare se un nodo è meta o no. Un algoritmo finale, f, che implementa la tecnica di ricerca considerata, fornisce una funzione che serve per ordinare gli elementi della lista OPEN al fine di determinare il prossimo nodo da estendere.

L'algoritmo generale è riassunto in forma di flowchart nella figura 5. Può essere descritto così:

1 Inserisci in OPEN il nodo iniziale. CLOSED è vuota.

2 Se OPEN è vuota non esistono soluzioni; il risultato è un insuccesso.

3 Altrimenti, poni A uguale al nodo in cima alla lista OPEN. Estrai A dalla lista OPEN e mettilo nella lista CLOSED.

4 Trova tutti i possibili successori di A, chiamali S(1), S(2),..., S(N).

5 Se non ci sono successori (N=0), va a **2**.

6 Controlla se i successori sono mete. Se c'è una meta, va a **8**.

7 Calcola i valori **f** di ogni successore ed inserisci i nodi nella lista OPEN così che i nodi di OPEN siano ordinati secondo valori **f** crescenti. Inoltre, poni ad ogni nodo successore un puntatore al nodo padre A. Va a **2**.

8 Risali dalla meta al nodo di partenza tramite i puntatori. Questa sequenza, letta al contrario, dà la soluzione.

9 Il gioco termina con successo.

Il metodo appena descritto è esaustivo e completo – cioè termina sempre. Il caso di "insuccesso" si ha quando tutti i nodi intermedi si sono estesi senza che venga trovata una meta.

Panoramica su RICERCA

Una spiegazione completa del programma RICERCA sarebbe troppo lunga, perciò mi limiterò ai punti necessari al lettore per ricomporre la meccanica del programma. Tutti i commenti sono riferiti alla versione a 16 caselle. (Ci si riporta al caso di 9 caselle ponendo 3 il valore R9 nell'istruzione 1800 del listato 1).

- Il programma del listato 1a è equivalente all'algoritmo in pseudolinguaggio del listato 1b ed al flowchart della figura 5.
- Le variabili principali del programma sono elencate nella tavola 1.
- Il cuore del programma RICERCA (che corrisponde all'algoritmo in pseudolinguaggio) va dalla riga 170 alla 610; le subroutine usate sono elencate nella tavola 2.
- I nuovi nodi, invece di essere ordinati all'interno della lista OPEN, vengono aggiunti alla fine e ad essi è assegnato un numero d'ordine. Invece di estendere il nodo in cima alla lista, viene esteso il nodo con numero d'ordine minore (tra quelli nel vettore O(N)). Così

listato 1a:

```
10 REM ***************************************************************
20 REM *                                                             *
30 REM *                    RICERCA SU UN ALBERO                     *
40 REM *                     PER IL GIOCO DEL 15                     *
50 REM *                                                             *
60 REM *                      PERSONAL SOFTWARE                      *
70 REM *                                                             *
80 REM ***************************************************************
85 CLEAR 1000: REM SOLO PER IL TRS-80
100 DIM O$(100),O(100),R$(20),A$(4),E$(4,4),F$(4,4),Z(16)
120 REM -- INIZIALIZZAZIONI ------------------------------------
130 GOSUB 1800
140 REM
150 REM -- CICLO DO-WHILE: FINCHE' "OPEN" NON
160 REM -- E' VUOTA, E EXIT="RICERCA"
170 IF E1$="USCITA" GOTO 660
180 REM
190 REM -- TROVA N1=INDICE DEL PIU' PICCOLO VALORE DI OPEN
210 GOSUB 950
220 IF O(N1)>=99999 GOTO 660
230 PRINT"230-ESPANSIONE NODO";N1;",";O$(N1);", VAL=";O(N1)
240 REM
250 REM -- INSERISCE IL NODO IN CLOSED
260 C9=C9+1
270 O(N1)=90000+O(N1)
280 REM
290 REM -- DECIFRA LO SCHEMA E GENERA I SUCCESSORI IN A$
310 E$=MID$(O$(N1),H1+1,L2)
320 E9$=MID$(O$(N1),H1,1)
330 GOSUB 1130
340 REM -- IL NODO N1 VIENE CONSIDERATO PER GENERARE I SUCCESSORI
360 GOSUB 1240
370 REM
380 REM -- CONTROLLO DELLE METE. SONO STATI GENERATI G1 NODI
400 IF G1>0 GOTO 430
410 PRINT"410-NESSUN SUCCESSORE PER IL NODO";N1
420 GOTO 610
430 G$="NON META": FOR M1=1 TO G1
440 REM -- SI OTTIENE META SE A$(M1) E' UNA META
450 GOSUB 2040
460 IF G$<>"META" GOTO 510
470 E1$="USCITA"
480 REM -- MEMORIZZA L'INDICE DELLA META
490 REM -- SALTO DALL'ISTRUZIONE 460
500 REM -- SCOMPONE LO SCHEMA NEI VETTORI E$ E F$
510 E$=MID$(A$(M1),H1+1,L2)
520 GOSUB 1130
530 REM -- CALCOLA H-CAPPELLO PER E$, RISULTATO IN R1
550 GOSUB 9900
560 REM -- PONE A$(M1) NELLA LISTA "OPEN"
570 O9=O9+1
580 O$(O9)=A$(M1)
590 O(O9)=R1
595 NEXT M1
600 REM -- FINE DEL CICLO DO-WHILE DELLA RIGA 170
610 GOTO 170
630 REM -----------------------------------------------------
640 REM -- STAMPA SOLUZIONE O MESSAGGIO DI INSUCCESSO
660 IF G$<>"META" GOTO 710
670 REM -- RICOSTRUISCE LA SOLUZIONE
680 GOSUB 2150
690 GOTO 720
700 REM -- NESSUNA SOLUZIONE
710 PRINT: PRINT"NON E' STATA TROVATA SOLUZIONE"
720 PRINT"NODI DELLA LISTA OPEN:  ";O9-C9
730 PRINT"NODI DELLA LISTA CLOSED:";C9
740 END
750 FOR I=1 TO R9
760 FOR J=1 TO R9
770 PRINT F$(I,J);: NEXT J
780 PRINT " ": NEXT I
790 RETURN
800 REM -----------------------------------------------------
810 REM -- CONTROLLO NUOVO SCHEMA F$(I,J)
820 Q3$="NON DOPPIO"
830 REM -- CONTROLLO SU TUTTI I NODI ESTESI, CIOE' QUELLI CON O(I)>=90000
850 FOR I=1 TO O9: IF O(I)<90000 GOTO 910
860 F1$=MID$(O$(I),H1+1,L2)
```

Listato 1. *Il programma RICERCA, scritto in Basic per il TRS-80. Il listato 1ª è il programma così come va implementato. Perché funzioni bisogna aggiungergli uno dei listati 2, 3 e 4. La subroutine che comincia alla riga 9900 implementa un certo metodo di ricerca; per ridurre le dimensioni del programma si possono togliere tutti i commenti (REM). Il messaggio diagnostico stampato dalla riga 230 dà un'idea retroattiva del funzionamento del programma, anche se non dà il valore corretto del nodo di partenza (nodo 1). Lo pseudolinguaggio del listato 1b chiarisce il funzionamento di RICERCA. In esso le etichette si riferiscono alle righe del programma.*

```
 870 IF F$=F1$ THEN Q3$="DOPPIO": RETURN
 880 NEXT I
 890 IF Q3$="NON DOPPIO" THEN RETURN
 900 PRINT"900--QUESTO SCHEMA E' DOPPIO ***"
 910 RETURN
 920 REM ------------------------------------------------------------
 930 REM -- RICERCA DELL'INDICE N1 PER CUI O(N1) E' MINIMO
 950 S1=99999: N1=1
 960 FOR I=1 TO O9
 970 IF O(I)>=S1 GOTO 990
 980 S1=O(I): N1=I
 990 NEXT I
1000 RETURN
1010 REM ------------------------------------------------------------
1020 REM -- RACCOGLIE LO SCHEMA F$(N,N) NELLA STRINGA F$
1040 F$=""
1050 FOR B=1 TO R9
1060 FOR D=1 TO R9
1070 F$=F$+F$(B,D)
1080 NEXT D
1090 NEXT B
1100 REM ------------------------------------------------------------
1110 REM -- SCOMPONE LA STRINGA E$ NELLE MATRICI E$(N,N) E F$(N,N),
1130 FOR I=1 TO R9
1140 FOR J=1 TO R9
1150 Q1-R9*(I-1)+J
1160 E$(I,J)=MID$(E$,Q1,1)
1170 F$(I,J)=MID$(E$,Q1,1)
1180 NEXT J: NEXT J
1190 RETURN
1200 REM ------------------------------------------------------------
1210 REM -- ESTENDE LA SITUAZIONE E$(I,J) NELLA DIREZIONE E9$,
1220 REM -- PONENDO I SUCCESSORI DEL NODO N1 IN A$(N), N=1,...,G1
1240 FOR I=1 TO 5
1250 IF E9$=MID$(D$,I,1) GOTO 1270
1260 NEXT I
1270 Q9$=MID$(I$,I,1)
1280 REM -- Q9$ E' LA DIREZIONE IN CUI NON SI PUO' ESTENDERE E$
1300 FOR Y1=1 TO R9: FOR X1=1 TO R9
1310 IF E$(X1,Y1)="." GOTO 1350
1320 NEXT X1: NEXT Y1
1330 REM -- X1,Y1=COORDINATE DELLO SPAZIO VUOTO NELLO SCHEMA
1350 G1=0
1360 REM -- IL PROSSIMO CICLO GENERA 4 POSSIBILI SUCCESSORI
1380 S1=1: A9=0
1390 IF S1>4 THEN RETURN
1400 IF MID$(D$,S1,1)=Q9$ GOTO 1610
1410 X2=X1+X(S1): Y2=Y1+Y(S1)
1420 FOR I=1 TO R9: FOR J=1 TO R9
1430 F$(I,J)=E$(I,J): NEXT J: NEXT I
1440 REM -- SCAMBIA LE POSIZIONI DI (X1,Y1) E (X2,Y2)
1460 IF X2<1 OR X2>R9 GOTO 1610
1470 IF Y2<1 OR Y2>R9 GOTO 1610
1480 F$(X1,Y1)=F$(X2,Y2): F$(X2,Y2)="."
1490 REM -- INSERISCE IL NUOVO NODO COME SCHEMA IN A$(N)
1510 GOSUB 1040
1520 REM -- CONTROLLO RIPETIZIONI IN O$
1530 GOSUB 820
1540 IF Q3$="DOPPIO" GOTO 1610
1550 REM -- NODO=PUNTATORE+DIREZIONE+VETTORE SCHEMA
1570 A9=A9+1
1580 GOSUB 1660
1590 A$(A9)=Q$+MID$(D$,S1,1)+F$
1600 G1=G1+1
1610 S1=S1+1: GOTO 1390
1620 RETURN
1630 REM ------------------------------------------------------------
1640 REM -- TRADUCE N1 IN STRINGA DI CARATTERI Q$
1660 Q$=STR$(N1)
1670 Q1=LEN(Q$)
1680 REM -- AGGIUNGE ZERI DAVANTI A Q$
1690 Q2=O8-Q1
1700 IF Q2>=0 GOTO 1730
1710 PRINT"ERRORE DI DIMENSIONE NELLA ROUTINE 1660"
1720 PRINT"... ABORTITO": END
1730 IF Q2=0 THEN RETURN
1740 FOR I=1 TO Q2
1750 Q$="0"+Q$: NEXT I
1760 RETURN
1770 REM ------------------------------------------------------------
1780 REM -- INIZIALIZZAZIONI
1800 O8=3: R9=3
1810 DATA -1,0,0,1,1,0,0,-1
1820 FOR I=1 TO 4: READ X(I),Y(I): NEXT I
1830 D$="BSADI":I$="ADBS"
1840 E9$="I"
1860 PRINT: PRINT"INSERISCI LO SCHEMA SOTTO FORMA DI";R9*R9;"CARATTERI"
1870 INPUT Q2$
1880 IF LEN(Q2$)=R9*R9 THEN 1910
1890 PRINT: PRINT"ERRORE NELL'INSERIMENTO DELLO SCHEMA"
1900 PRINT"RIBATTI": GOTO 1860
1910 O9=1: O(1)=0
1920 N1=1: GOSUB 1660
1930 O$(1)=Q$+"I"+Q2$
1940 REM INIZIALIZZATE LE LISTE "OPEN", VETTORI O, O$
```

si risparmia al programma un ordinamento superfluo.

- Si può anche far senza il vettore CLOSED, contrassegnando con un valore di O maggiore di 90000 i nodi già estesi. Il numero di nodi già estesi è dato da C9.

- Nel programma RICERCA certe costanti sono inserite in variabili, cosicché si possono facilmente apportare modifiche per l'adattamento del programma a computer di dimensioni diverse. È il caso delle variabili L2, O8 ed R9 della tavola 1.

- La rappresentazione completa di un nodo nel programma è data dalle variabili O(N) e O$(N), dove N è il numero dei nodi. O(N) è il numero d'ordine relativo agli altri nodi; O$(N) contiene tre informazioni: la direzione A (alto), B (basso), D (destra), S (sinistra) secondo cui si arriva dal predecessore di O$(N) ad O$(N) stesso, un puntatore di O8 cifre che dà il numero di nodo del predecessore di O$(N), ed una stringa di L2 caratteri che riassume in forma sintetica lo "schema" corrispondente al nodo N. (Per ulteriori dettagli si veda la figura 6.)

- Per ogni nodo, diverso da quello di partenza, è sempre possibile eliminare una delle "mosse" consentite. Ad esempio, se O(N1) genera O(N2) con la mossa B (basso), possiamo ignorare l'ipotetico successore di O(N2) generato dallo spostamento A (alto), perché si otterrebbe come risultato lo stesso schema del nodo N1 (che è già stato esteso). Nel programma RICERCA questa operazione si compie confrontando la "direzione" del nodo corrente, contenuta in D$, con la lettera corrispondente nella variabile I$ (I$ sta per "Inverso"); quest'ultima viene quindi ignorata nell'estendere il nodo in questione (vedi la subroutine 1240).

- Può darsi che una serie di mosse riporti ad un nodo già precedentemente esteso (ciò significa che lo spazio degli stati per il gioco del 15 è un grafo, non un albero). Per questo motivo si controlla se i successori generati nelle righe da 1360 a 1610 (subroutine 1240) sono

già stati considerati (subroutine 820, richiamata alla riga 1530).

- Ogni mossa possibile nel gioco del 15 è associata ad un numero da uno a quattro (1 rappresenta lo spostamento in giù della piastrina, 2 lo spostamento a sinistra, 3 in su, 4 a destra). Le variabili che si servono di questa numerazione sono X(N), Y(N), D$ e I$.
- Nella rappresentazione dello schema con caratteri, il punto simboleggia la posizione dello spazio vuoto.
- Il programma RICERCA è stato provato su un TRS-80 modello I e dovrebbe funzionare senza modifiche su PET Commodore, Radio Shack TRS-80 modello III, Apple II e su qualsiasi altro computer che utilizzi la versione Microsoft del Basic. Tutte le righe di commento (REM) possono essere tolte senza che le prestazioni del programma vengano alterate.

```
1950 REM INIZIALIZZAZIONE LISTE "CLOSED". C. C$ VUOTI
1960 C9=0
1970 H1=O8+1: L2=R9*R9
1980 E1$="RICERCA"
1990 G$="NON META"
2000 RETURN
2010 REM ----------------------
2020 REM -- CONTROLLA SE E' UNA META
2030 REM -- RESTITUISCE G$="META" O "NON META"
2040 Q3$="12345678."
2050 IF R9=4 THEN Q3$="123456789ABCDEF."
2060 Q4$=RIGHT$(A$(M1),L2)
2070 IF Q3$-Q4$ THEN G$="META": N6=O9+1
2080 REM -- N6=INDICE DI META=O9+1 PERCHE' O9 PUNTA
2090 REM -- ALL'ULTIMO NODO CONSIDERATO: VEDI 560-580
2100 RETURN
2120 REM -----------------------------------------------------
2130 REM -- RICOSTRUISCE LA SOLUZIONE DA O$(N1)
2150 R1-0
2160 REM -- FINCHE' IL PUNTATORE="I"
2170 O$=MID$(O$(N6),O8+1,1)
2180 IF Q$="I" THEN 2200
2190 R1=R1+1: R$(R1)=O$
2200 Q1$=LEFT$(O$(N6),O8)
2210 N6=VAL(Q1$)
2220 IF Q$<>"I" THEN 2170
2230 REM
2240 IF R1<=0 THEN PRINT: PRINT"ERRORE ZERO NEL BACKTRACK": RETURN
2250 PRINT: PRINT"LA SOLUZIONE E'";
2260 FOR N=R1 TO 1 STEP -1
2270 PRINT" ";R$(N);
2280 NEXT N
2290 PRINT: PRINT: PRINT R1;" PASS";
2300 IF R1>1 THEN PRINT"I": GOTO 2310
2305 PRINT"O"
2310 PRINT: PRINT
2320 RETURN
```

listato 1b:

```
130     EXIT=no, META=no, routine d'inizializzazione
        OPEN=nodo iniziale

170     do while OPEN non vuota e EXIT=no
210         : A=nodo in cima a OPEN
260         : sposta A da open a CLOSED
360         : estende A ai successori A(N)

400         : if ci sono successori
430         :   : for ogni successore
450         :   :    : controlla se è meta
460         :   :    : if il nodo è meta
470         :   :    :   : poni G=indice del nodo, META=sì
            :   :    :   : poni EXIT=sì
            :   :    :   : else

550         :   :    :   : calcola il "valore" del nodo
570         :   :    :   : inserisci il nodo col valore in OPEN
            :   :    : end if
595         :   :    : end del ciclo for
            :   : end if
610         : end while

660     if META=sì
680         : ricostruisce la soluzione a partire da G
            : stampa la soluzione
        else
710         : stampa messaggio di errore
        end if
740     end program
```

I metodi di ricerca sistematica sono sostanzialmente tre

Un primo metodo di ricerca sistematica su un albero può essere descritto così: estendere il nodo iniziale, memorizzando tutti i successori (che stanno a livello 1); se nessun nodo è una meta, si estendono tutti i nodi del livello 1 per ottenere quelli di livello 2; si ripete il procedimento finché si trova una meta o finché non si trovano più nodi da estendere.

In termini dell'algoritmo principale di ricerca (figura 5), questo metodo detto *breadth-first* viene implementato ponendo i successori appena generati in cima alla lista OPEN – oppure, equivalentemente, assegnando loro un valore O(N) uguale al livello in cui si trovano. La figura 7a mostra l'ordine in cui i nodi vengono estesi secondo la ricerca *breadth-first*; si noti che tutti i nodi di livello n vengono estesi prima di qualsiasi nodo di livello $(n+1)$. Il listato 2 è quello

Su AMITALIA il sole splende ALTOS, i nuovi microcomputers "anni luce" avanti. Su tutti.

* **CP/M, MP/M** sono marchi registrati della Digital Research.
* **OASIS** è un marchio registrato della Phase One.

AMITALIA, rappresenta in esclusiva per il mercato italiano una grande famiglia di microcomputers su singola scheda da 8 e 16 bit: gli ALTOS, protagonisti della microinformatica più avanzata, risultati di una tecnologia che viene dal domani per tutte le esigenze di mono e multiutenza di oggi. Microcalcolatori, gli ALTOS, che ricordano e parlano meglio di ogni altro tutte le lingue dell'informatica distribuita.
AMITALIA è anche un'organizzazione leader di distribuzione e assistenza che copre, con personale qualificato e specialistico, l'intero territorio nazionale.
Ma passiamo a conoscerli meglio tecnicamente questi microcomputers "anni luce" avanti su tutti.

ACS 8000
MICROPROCESSORE 8 BIT
SUPPORTO DI MEMORIA 8"
FLOPPY E HARD DISK
RICOVERO DATI SU CASSETTA MAGNETICA

da 64 K RAM di memoria a 208 K RAM di memoria
Floppy disk singola faccia doppia densità 0,5 MByte
Dischi fissi da 10, 20, 40, 80 MByte in linea
Cassetta magnetica per ricovero dati da 17,5 MByte
da 1 a 4 terminali per multiutenza
Sistemi operativi: *CP/M, *MP/M, *OASIS

ACS 5
MICROPROCESSORE A 8 BIT
SUPPORTO DI MEMORIA 5 1/4"
FLOPPY E HARD DISK

196 K RAM di memoria
Floppy disk doppia faccia doppia densità 1 MByte
Dischi fissi da 5, 10, 20 MByte in linea
da 1 a 3 terminali per multiutenza
Sistemi operativi: *CP/M, *MP/M, *OASIS

ALTOS
COMPUTER SYSTEMS

ACS 8600
MICROPROCESSORE A 16 BIT
SUPPORTO DI MEMORIA 8"
FLOPPY E HARD DISK
RICOVERO DATI SU CASSETTA MAGNETICA

da 500 a 1000 K RAM di memoria
Floppy disk singola faccia doppia densità 0,5 MByte
Dischi fissi da 10, 20, 40, 80 MByte in linea
Cassetta magnetica per ricovero dati da 17,5 MByte
da 1 a 8 terminali per multiutenza
Sistemi operativi: *CP/M-86, *MP/M-86, *OASIS-16, XENIX

AMITALIA, SAICO, SEGI: tre leader un gruppo, AMMI.

AMITALIA
ADVANCED MICROCOMPUTER ITALIA s.r.l.
20124 Milano - Via Volturno, 46 - Tel. (02) 683985 - 6881946 - 6898015
00142 Roma - Via B. Croce, 97 - Tel. (06) 5410620

Nome della variabile	Uso
A$(N)	Vettore dei successori generati dalla situazione attuale dello schema E$(I,J); vedi A9.
F$(N)	Stringa di caratteri corrispondente allo schema F$(I,J); vedi la subroutine alla riga 1040.
O(N)	Numero d'ordine del nodo N nella lista OPEN; il nodo appartiene alla lista CLOSED se O(N)>90000.
O$(N)	Dati del nodo N della lista OPEN; vedi il testo.
R$(N)	Lettere che, lette in ordine inverso, danno la soluzione del gioco; vedi R1.
X(N),Y(N)	Incrementi dalle posizioni x ed y che producono lo spostamento nella direzione N, N=1,2,3,4.
E$(I,J)	Rappresentazione del nodo attuale in forma estesa; I e J variano tra 1 e R9.
F$(I,J)	Schema ausiliario che serve per generare i successori dello schema E$(I,J).
A9	Numero di successori generati senza ripetizioni; vedi A$(N).
C9	Numero di nodi segnati come già estesi, vedi O(N).
D$	I caratteri di D$ sono i vari movimenti possibili (in basso, a sinistra, in alto, a destra) in ordine (cioè mossa 2=MID$(D$,2)="S"=sinistra; la mossa 5, I, significa "inizio" e si applica solo al nodo iniziale).
E9$	Direzione usata per ottenere il nodo attuale a partire dal suo predecessore.
G$	Indica se A$ (M1) è meta o no.
G1	Numero di successori generati prima che si controlli se ci sono ripetizioni.
H1	(Indice del primo carattere di gioco di O$(N)) meno 1; serve per elencare gli L2 caratteri dello schema.
L2	Numero di caratteri nello schema corrente; =9 per lo schema di ordine 3,=16 per lo schema di ordine 4.
N1	Indice di O(N) che dà il più piccolo numero d'ordine; il nodo N1 sarà il prossimo ad essere esteso.
O7	Dimensione massima dei vettori O(N) e O$(N).
O8	Numero di cifre del puntatore di O$(N) al suo predecessore; è fissato a 3, ma può essere ampliato per consentire ricerche più complesse.
O9	Numero dei nodi (estesi o non estesi) nella lista O$(N); i nuovi nodi seguenti saranno posti in O$(O9+1).
O9$	Direzione di backtrack dal nodo attuale al predecessore; vedi il testo.
R1	Numero di passi per ottenere la soluzione; vedi R$(N). Inoltre, valore prodotto dalla subroutine 9900.
R9	Ordine del problema; uguale a 3 per lo schema a 9 caselle ed a 4 per lo schema a 16 caselle.
S1	Minimo numero d'ordine nel vettore O(N), S1=O(N1).

Tav. 1. *Principali variabili del programma Basic RICERCA (vedi listato 1).*

della subroutine che implementa questo algoritmo; tradotta in pseudolinguaggio è:

```
9900 valore del nodo ritornato,
     R1=valore del padre
     O(N1)+1
```

Nella ricerca *breadth-first*, ponendo il "valore" di un nodo uguale al suo livello si ha che, prima di qualsiasi nodo di livello $n+1$ (con "valore" $n+1$), vengono estesi tutti quelli al livello n. Nella riga 9900 viene tolto il valore 90000, perché il nodo padre O(N1) è stato "segnato" come già esteso aggiungendogli appunto 90000.

Un altro modo di affrontare la ricerca è l'*algoritmo depth-first*. In esso si estendono ripetutamente i successori di un dato nodo (finché nessuno dei nodi ottenuti può essere esteso) prima di cominciare ad estendere un altro modo dello stesso livello; insomma, si percorre l'albero dall'alto in basso invece che in orizzontale e, in sostanza, i nodi terminali vengono generati ed estesi da sinistra verso destra. Nella figura 7b è illustrato l'ordine in cui i nodi di un albero vengono estesi nella ricerca depth-first; la subroutine da inserire in RICERCA è quella del listato 3. In pseudolinguaggio corrisponde a

```
9900 valore del nodo ritornato,
     R1=valore del padre
     O(N1)−1
```

In una ricerca *depth-first* pura, i nodi appena generati dovrebbero essere estesi prima di qualsiasi altro nodo appartenente alla lista OPEN già da estensioni precedenti. Una soluzione consiste nel porre il "valore" del successore uguale a quello del padre meno 1. Poiché le subroutine 9900 sceglie il nodo con valore minimo nel vettore O, l'ordine in cui si effettuano le estensioni viene modificato.

Seguendo l'algoritmo *depth-first*, si ha l'impressione che generi nodi

sinclair

è l'unico sistema completo a 550.000 lire.

ZX81 8 k ROM, 1 k RAM	L. 199.000
ESPANSIONE 16 k RAM	L. 131.000
STAMPANTE ZX	L. 220.000

Sinclair, sempre Sinclair: poco più di mezzo milione per un completo sistema di computing.
Guarda, confronta, cerca un'alternativa!
A questo prezzo non trovi neanche un'unità centrale: figuriamoci poi 16 k e la stampante.
Oggi più che mai la chiave che apre le porte dell'informatica per tutti è Sinclair.

REBIT COMPUTER
A DIVISION OF G.B.C.

Prezzi IVA esclusa.

Per informazioni scrivere a CASELLA POSTALE 10488 - MILANO

Etichetta	Uso
820	Input: Schema F$(I,J), vettore O$(N). Calcolo: Si controlla se lo schema F$(I,J) è già stato esteso. Output: Q3$="DOPPIO" o "NON DOPPIO".
950	Input: Vettore O$(N), O9. Calcolo: Calcolo del minimo nel vettore. Output: Indice N1, valore S1 tale che S1=O(N1) è il minimo in O(N).
1040	Input: Schema F$(I,J). Calcolo: Riduzione dello schema a stringa di caratteri. Output: F$=Stringa di L2 caratteri.
1130	Input: Stringa E$. Calcolo: Scompone la stringa negli schemi E$(I,J) e F$(I,J). Output: Schemi E$(I,J) e F$(I,J).
1240	Input: Schema E$(I,J), numero di nodi N1, direzione E9$. Calcolo: Genera fino a tre successori dello schema (derivato dal nodo N1), elimina i nodi già estesi, completa i dati di ogni successore (direzione+puntatore+schema). Output: Tavola dei successori A$(N), A9.
1660	Input: Numero N1, lunghezza voluta O8. Calcolo: Traduce N1 in stringa; aggiunge gli zeri davanti. Output: Stringa "uguale"Q$ ad N1.
1800	Routine di inizializzazione; input schema da risolvere.
2040	Input: Nodo A$(M1), ordine del problema R9. Calcolo: Riduce il nodo in forma di stringa e controlla se è meta o no. Output: G$="META" o "NON META".
2150	Input: Meta O$(N1). Calcolo: Ricostruzione della soluzione tramite i puntatori e raccolta delle "direzioni" in R$(N). Output: Stampa la soluzione (R$(R1), R$(R1−1),...,R$(1)).
9900	Input: Nodo O$(M1) da aggiungere alla lista, lista O(N) (ed altre variabili a seconda del metodo). Calcolo: Arrangiamento dei valori in O(N); calcolo del valore d'ordine del nodo M1, messo in R1, in modo che il nodo M1 venga inserito nella lista OPEN con un numero d'ordine corretto. Output: Subroutine che implementa un certo algoritmo di ricerca.

Tav. 2. *Descrizione delle subroutine del programma RICERCA.*

senza mai fermarsi – ma senza tentare di avvicinarsi ad una meta (se non quando ci si trova all'estrema sinistra dell'albero). Lo svantaggio del metodo *depth-first* è che, prima di tornare a livelli inferiori, si deve arrivare alla fine di un ramo. Siccome gli alberi di solito sono molto estesi, se non infiniti, (e spesso esiste una meta a livelli bassi), l'algoritmo *depth-first* è generalmente meno efficiente del metodo *breadth-first*: il primo scandisce su e giù l'albero in tutta la sua lunghezza, da sinistra a destra, mentre l'altro percorre ogni livello dalla cima al fondo dell'albero, fermandosi se trova una meta a livello intermedio.

L'*algoritmo depth-first limitato* (figura 7c e listato 4) si può riassu-

Listato 2: *Routine che implementa la strategia di ricerca breadth-first.*

```
9890 --RICERCA BREADTH-FIRST
9896 --RISULTATO R1=VALORE DEL PADRE+1
9900 R1=(O(N1)+1)-90000
9905 RETURN
```

Listato 3: *Routine che implementa una ricerca depth-first pura.*

```
9890 --RICERCA DEPTH-FIRST PURA
9900 R1=O(N1)-90000)-1
9905 RETURN
```

Listato 4: *Routine che implementa una ricerca depth-first limitata.*

```
9890 --RICERCA DEPTH-FIRST LIMITATA
9900 R3=3
9905 R1=(O(N1)-90000)-1
9910 IF R1=-R3 THEN R1=0
9915 RETURN
```

mere come segue. Scegliamo un livello arbitrario *n*; si effettua la ricerca *depth-first* non considerando (per il momento) tutti i nodi di livello maggiore o uguale ad *n*. Se non si trova una meta, la ricerca è trasferita ai nodi precedentemente ignorati, tralasciando quelli di livello maggiore o uguale a 2*n*. Si ripete il procedimento finché si trova una meta oppure si è percorso tutto l'albero. Tutto si può riassumere in pseudolinguaggio così:

9900 R3=livello di ogni "fascia" di ricerca
9905 valore del nodo ottenuto, R1=valore del padre O(N1)−1
9910 se il "valore" è uguale a R3 riportarlo a zero.

(6a)

NODO 1

NODO 10

NODO 23

(6b)

O(10) = 5 O$(10) = "S 001 12345.786"
O(23) = 3 O$(23) = "A 010 12345678."

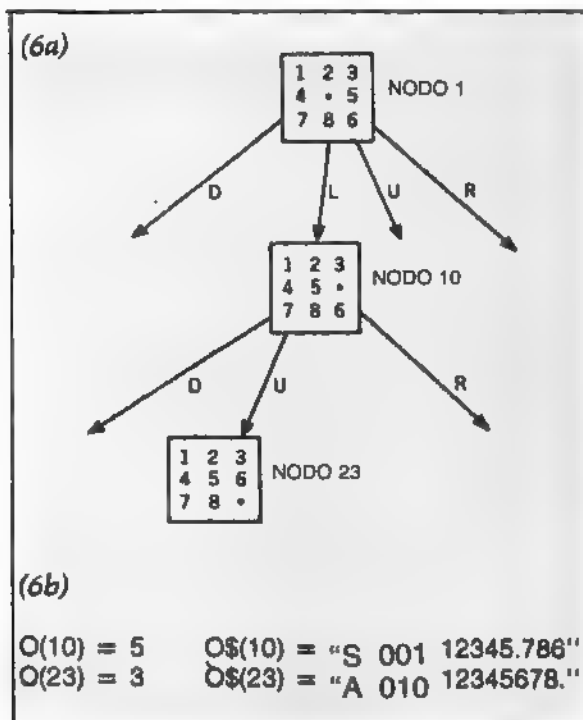



Fig. 6. *La rappresentazione del gioco dell'8 all'interno del programma RICERCA. La condizione di ogni nodo è definita da due variabili. La prima è O(N), il numero d'ordine del nodo N dell'albero. L'altra è O$(N), una stringa composta da:*
- *la direzione seguita per ottenere N dal nodo padre.*
- *il numero di nodo del padre di N (serve per collegare N al nodo di partenza)*
- *la situazione dello schema relativo al nodo N, scritta per riga.*

La figura 6a mostra parte di un ipotetico albero; la figura 6b invece dà la rappresentazione dei nodi 10 e 23 all'interno del programma.

(7a) (7b) (7c)

LIVELLO

n = 2

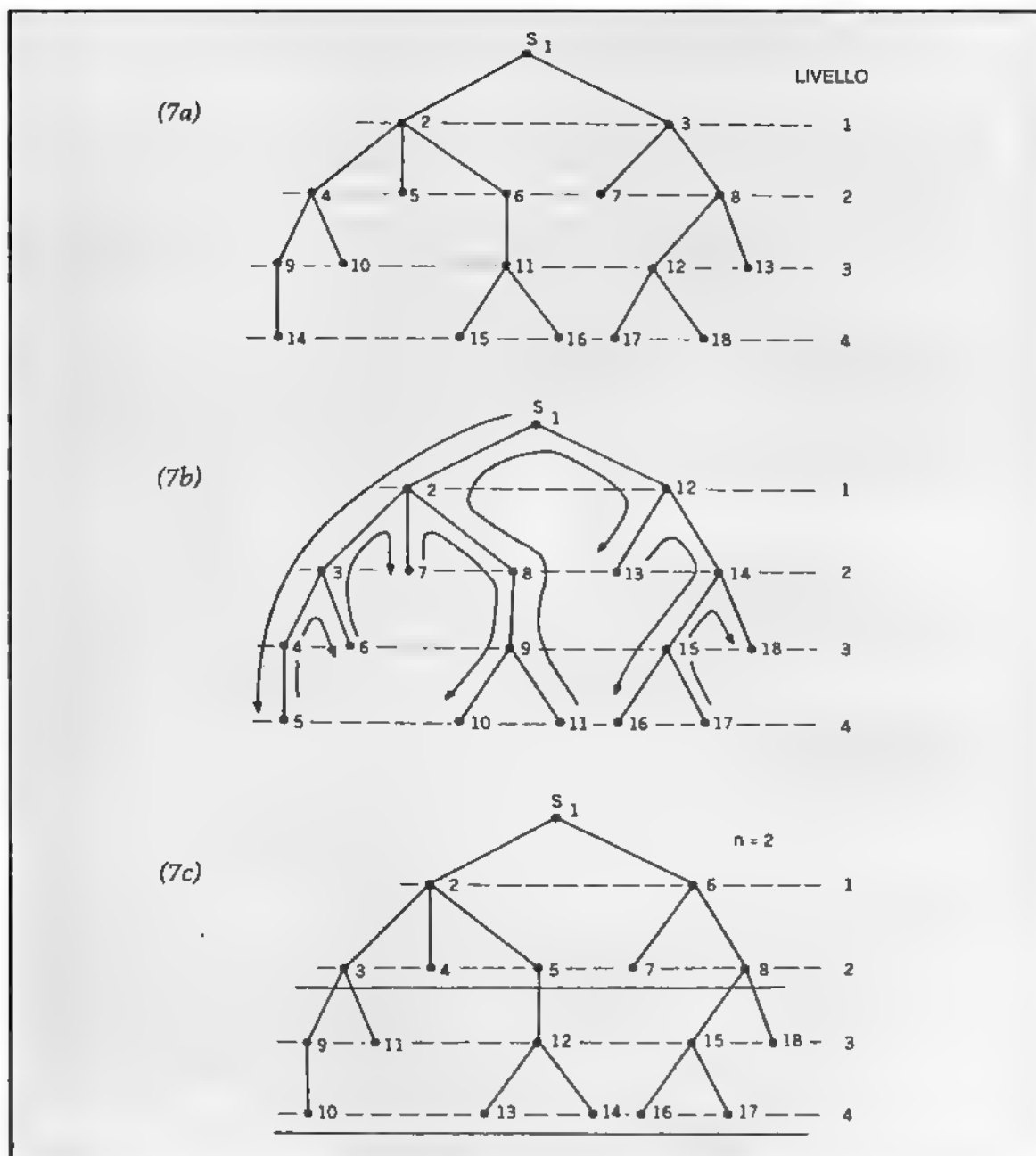



Fig. 7. *Ordine di estensione secondo tre diversi algoritmi esaustivi di ricerca. I numeri alla destra di ogni modo suggeriscono l'ordine in cui l'albero si estende. Nella figura 7a, secondo l'algoritmo breadth-first, la ricerca si sviluppa in orizzontale, un livello dopo l'altro; nella figura 7b, la ricerca depth-first percorre l'albero nel senso indicato dalle frecce; la ricerca depth-first limitata, in figura 7c, combina le proprietà degli altri due metodi. Questi algoritmi di ricerca possono essere provati inserendo le routine dei listati 2, 3 o 4 nel programma RICERCA (listato 1).*

Il "valore" da ritornare al programma sarà perciò zero o negativo. Questa ricerca è limitata di volta in volta al livello R3 ponendo un nodo a lato del livello corrente al valore più alto possibile, zero, così che sarà esteso solo quando la fascia in questione si sarà estesa fino al limite. Il valore R3 può essere fissato nella routine d'inizializzazione 9500. La ricerca depth-first limitata allevia gli svantaggi del metodo *depth-first* puro, perché effettua una ricerca eventualmente esaustiva solo a fascie di livelli. Ma si tratta di un compromesso: se c'è una meta in profondità e nella parte sinistra dell'albero, la si troverebbe prima con la ricerca *depth-first pura*. (In generale, l'efficienza delle due tecniche *depth-first* dipende in grande misura dalla posizione orizzontale delle mete nell'albero.)

Appunti sulle prove

Ora abbiamo gli strumenti per analizzare gli algoritmi esaustivi di ricerca. Il programma Basic RICERCA (completato con l'appro-

priata subroutine 9900) richiede in input lo schema iniziale (letto riga per riga, con le lettere da A ad E per rappresentare le cifre da 10 a 15 nella versione di ordine 4, ed il punto decimale al posto dello spazio vuoto), elenca i nodi quando vengono generati ed estesi, e stampa la soluzione del gioco. Se volete evitare di inserire uno schema ad ogni esecuzione del programma, potete aggiungere

```
1850 Q2$="<schema in input>":
1855 GOTO 1910
```

Le mie osservazioni e le mie prove sono basate sul campione di gioco dell'8 illustrato in figura 8; in essa uno schema è definito dai suoi numeri di riga e di colonna. Si noti che il numero di riga corrisponde al numero di mosse sufficienti per risolvere il problema e che ogni schema è un sottoproblema di quello che sta sotto nella stessa colonna.

La tavola 3 fornisce i dati relativi ad alcuni schemi ed agli algoritmi *breadth-first* e *depth-first* limitato.

Ricordate che i nodi di CLOSED sono quelli già estesi ai successori, mentre quelli di OPEN sono quelli generati ma non ancora estesi; perciò, il numero totale di nodi generati da un algoritmo è dato dalla somma di quelli delle due liste.

Si tenga presente pure che i successori vengono generati per estensione nel seguente ordine: in basso, a sinistra, in alto, a destra. Ciò è molto importante per le ricerche esaustive, ma lo è meno in altri casi.

(3a)

Schema	Soluzione	El.Open	El.Closed	Totale
(1,1)	S	3	1	4
(1,2)	A	3	1	4
(2,1)	AS	7	4	11
(2,2)	SA	6	3	9
(2,3)	SS	4	3	7
(3,1)	SAS	9	8	17
(3,2)	ASA	7	6	13
(3,3)	ASS	10	9	19
(4,1)	BSAS	12	11	23
(4,2)	SASA	10	9	19
(4,3)	DASS	16	21	37

(3b)

Schema	Soluzione	El.Open	El.Closed	Totale
(1,1)	S	3	1	4
(1,2)	A	3	1	4
(2,1)	AS	11	12	23
(2,2)	SA	8	7	15
(2,3)	SS	10	9	19
(3,1)	SAS	11	13	24
(3,2)	ASA	9	7	16
(3,3)	ASS	12	17	29
(4,1)	BSAS	9	8	17
(4,2)	SASA	7	6	13
(4,3)	DASS	16	21	37

(3c)

Schema	Soluzione	Livello (n)	El.Open	El. Closed	Totale
(2,2)	SA	2	6	3	9
		3	6	5	11
		4	8	7	15
		5	12	11	23
(2,3)	SS	2	4	3	7
		3	6	5	11
		4	10	9	19
		5	12	17	29

Tav. 3. *Prove di RICERCA. Nella ricerca breadth-first (3a), "El.Open" è il numero di nodi creati ma non ancora estesi. "El.Closed" è il numero di nodi già estesi. Poiché l'algoritmo breadth-first effettua la ricerca uniformemente a partire dal nodo iniziale, è il più affidabile tra i metodi esaustivi di ricerca. Nella ricerca depth-first limitata (3b), con limite n=4, per tutti gli schemi al livello quattro il numero di nodi generati è minore o uguale che nella ricerca breadth-first. Variando il limite n, l'algoritmo effettua la ricerca a "fascie" (3c), ognuna di profondità n. Quando n diventa maggiore del numero di mosse della soluzione (in questo caso due), l'efficienza dell'algoritmo si accosta a quella del metodo depth-first.*

Osservazioni e discussione

Cercate di rispondere alle seguenti domande prima di consultare le risposte.

A proposito della ricerca *breadth-first*:

- Avrete già notato che il numero di nodi estesi per ottenere una soluzione varia a seconda delle direzioni scelte; le mosse "in basso" e "a sinistra" tendono a rendere più rapida la soluzione, mentre gli

La prima e la più grande catena di computer in Italia: ogni volta che ci vai, c'è sempre qualcosa di nuovo.

Tandy

BMC

DAI THE MICROCOMPUTER COMPANY

VIC-20

Honeywell

sinclair

ARFON MICRO

SHOP bit primavera

CASIO

ATARI®

SONY®

GRUPPO EDITORIALE JACKSON

SAMSUNG

SEIKOSHA

ALESSANDRIA Via Savonarola, 13 • **AREZZO** Via F. Lippi, 13 • **BARI** Via Capruzzi, 192 • **BERGAMO** Via F. D'Assisi, 5 • **BOLOGNA** Via Brugnoli, 1 A • **CAGLIARI** Via Zagabria, 47 60 • **CAMPOBASSO** Via Mons. II Bologna, 10 • **CESANO MADERNO** Via Ferrini, 6 • **COMO** Via L. Sacco, 3 • **COSENZA** Via dei Mille, 86 • **FAVRIA CANAVESE** C.so Matteotti, 13 • **GALLARATE** Via A. Da Brescia, 2 • **L'AQUILA** Via Strada 85, 2 • **MESSINA** Via Del Vespro, 71 • **MILANO** Galleria Manzoni, 40 • **MILANO** Via Petrella, 6 • **MILANO** Via G. Cantoni • **MILANO** P.zza Firenze, 4 • **MILANO** Via Altaguardia, 2 • **MILANO** V.le Corsica, 14 • **NOVARA** Via Q. Sella, 32 • **PADOVA** Via Fistomba, 8 • **PALERMO** Via Lamarmora, 82 • **PARMA** Via Borghesi, 16 • **PAVIA** Via C. Battisti, 4 A • **PESCARA** Via Guelfi, 74 • **PISA** Via XXIV Maggio, 101 • **PISTOIA** V.le Adua, 350 • **POZZUOLI** Via Pergolesi, 13 • **RIMINI** Via Bertola, 75 • **ROMA** Via Cerreto Da Spoleto, 23 • **ROMA** P.zza San Dona di Piave, 14 • **SONDRIO** Via Nazario Sauro, 28 • **TERAMO** Via Martiri Pennesi, 14 • **TERNI** Via P. Gori, 8 • **TORINO** Via Chivasso, 11 • **TORINO** C.so Grosseto, 209 • **TORINO** Via Tripoli, 179 • **TRIESTE** Via F. Severo, 138 • **VARESE** Via Carrobbio, 13 • **VERONA** Via Pontiere, 2 • **VIAREGGIO** Via Volta, 79 • **VOGHERA** P.zza Carducci, 11

In tutta Italia, con più computer.

spostamenti "in alto" e "a destra" la allungano. Confrontate lo schema (2,3) con (2,1) e (4,2) con (4,3).

• (Domanda 1) Come si riflette questa tendenza sul numero di nodi della lista CLOSED? (Le soluzioni seguono nel riquadro.)

• (Domanda 2) Come mai lo schema (3,1), con soluzione SAS, dà luogo a più nodi di (3,2), che ha soluzione ASA? Le mosse S non si estendono prima delle mosse A?

• Si noti che i due schemi (1,1) e (1,2) hanno lo stesso numero di nodi in OPEN e CLOSED, sebbene la soluzione S di (1,1) preceda quella di (1,2), A. Questo succede perché il programma RICERCA genera *tutti* i successori validi nel nodo che viene esteso, prima di controllare se sono mete.

• (Domanda 3) Esaminando la soluzione di due schemi, di cui uno è sottoproblema dell'altro, si può notare un fatto curioso: il sottoproblema genera un nodo non prodotto dal problema più ampio. Trovate le soluzioni degli schemi (2,1) e (3,1) con la subroutine breadth-first ed elencate i nodi generati. Qual è il nodo in questione? Perché discende da (2,1) e non da (3,1)?

Sulla ricerca *depth-first*:

• Troverete anche voi che, prima di trovare una soluzione con il metodo depth-first, è facile superare i limiti di memoria (e di pazienza). Questo perché bisogna discendere fino in fondo all'albero, che è molto alto anche per problemi piccoli come il gioco dell'8 in ordine 3. Per farvi un'idea, provate a risolvere col programma RICERCA e l'algoritmo depth-first il problema di ordine 2.

Per quanto riguarda la ricerca *depth-first* limitata:

• La tavola 3b illustra i risultati della soluzione degli stessi schemi, (2,2) e (2,3), usando limiti diversi (la variabile D3 della riga 9900). Poiché entrambi gli schemi vengono risolti in due mosse, il limite determina semplicemente di quanto bisogna scendere con la ricerca prima di ritornare a livelli inferiori.

• Per definizione una ricerca depth-first limitata al livello *n* si estende *n*−1 livelli alla volta, generando i nodi ma non estendendoli immediatamente. Dunque questo metodo è più efficiente quando il livello della ricerca è uguale al numero di mosse della soluzione: la ricerca scende per la parte sinistra dell'albero, si ferma al livello *n*, e scandisce il sottoalbero del livello n da sinistra a destra.

Meta	1 2 3 4 5 6 7 8 .		
Numero di riga			
(1) Ad una mossa	1 2 3 4 5 6 7 . 8	1 2 3 4 5 . 7 8 6	
(2) A due mosse	1 2 3 4 . 6 7 5 8	1 2 3 4 5 7 8 6	1 2 3 4 5 6 7 8
(3) A tre mosse	1 2 3 . 4 6 7 5 8	1 3 4 2 5 7 8 6	1 2 3 5 6 4 7 8
(4) A quattro mosse	1 2 3 7 4 6 5 8	. 1 3 4 2 5 7 8 6	1 2 3 5 6 4 7 8

Fig. 8. *Tavola di rappresentazione del gioco dell'8, usata nel testo. Ignorando la riga della meta, ogni schema è definito dai numeri di riga e di colonna. P.es. lo schema più a destra nella riga "a due mosse" è chiamato (2,3). Si noti la relazione tra gli schemi della stessa colonna: ogni schema è un sottoproblema di quello immediatamente sotto.*

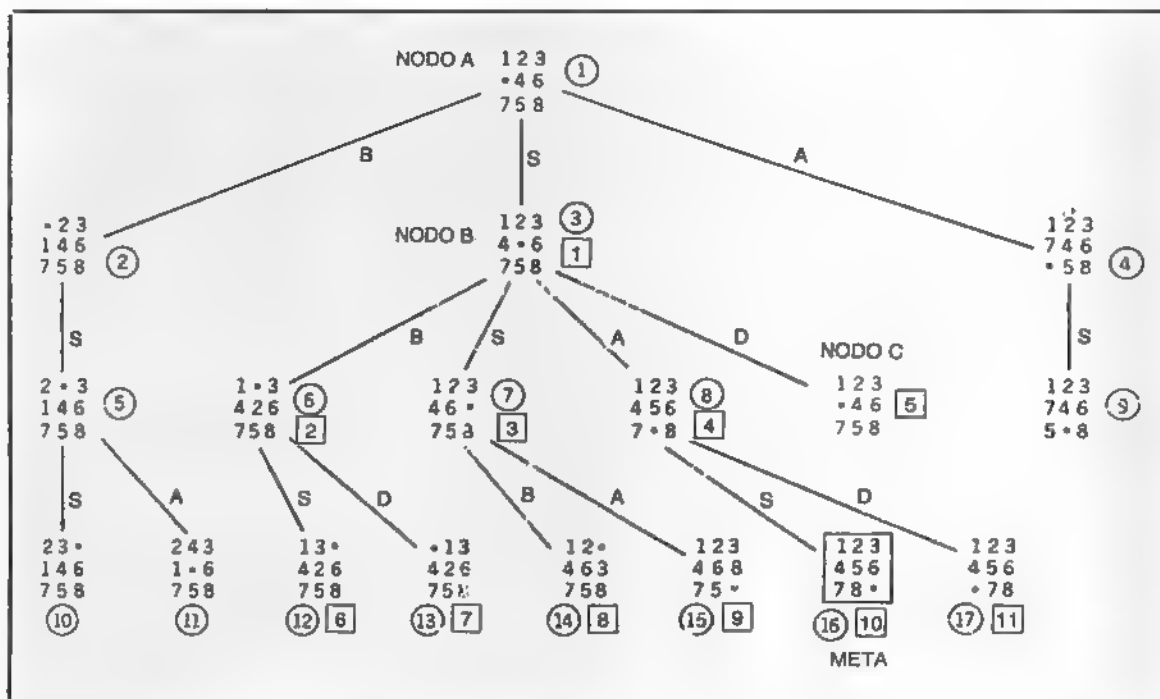



Fig. 9. *Soluzione dello schema (3,1) con l'algoritmo breadth-first. L'albero è stato sviluppato solo fino a raggiungere una meta (contrassegnata nella riga in fondo). I numeri nel cerchietto nero suggeriscono l'ordine in cui i nodi vengono estesi per risolvere il gioco. Il nodo B è il nodo di partenza dello schema (2,1), sottoproblema dello schema (3,1); i numeri nei quadratini indicano invece l'ordine in cui i nodi si estendono per risolvere (2,1).*

Fig. 10. *Soluzione dello schema (3,2) con l'algoritmo breadth-first. I numeri nel cerchietto a destra di ogni nodo indicano l'ordine in cui vengono estesi. I nodi da 8 a 13 (sei nodi) sono nella lista OPEN (cioè non sono ancora estesi), mentre quelli da 1 a 7 appartengono a CLOSED.*

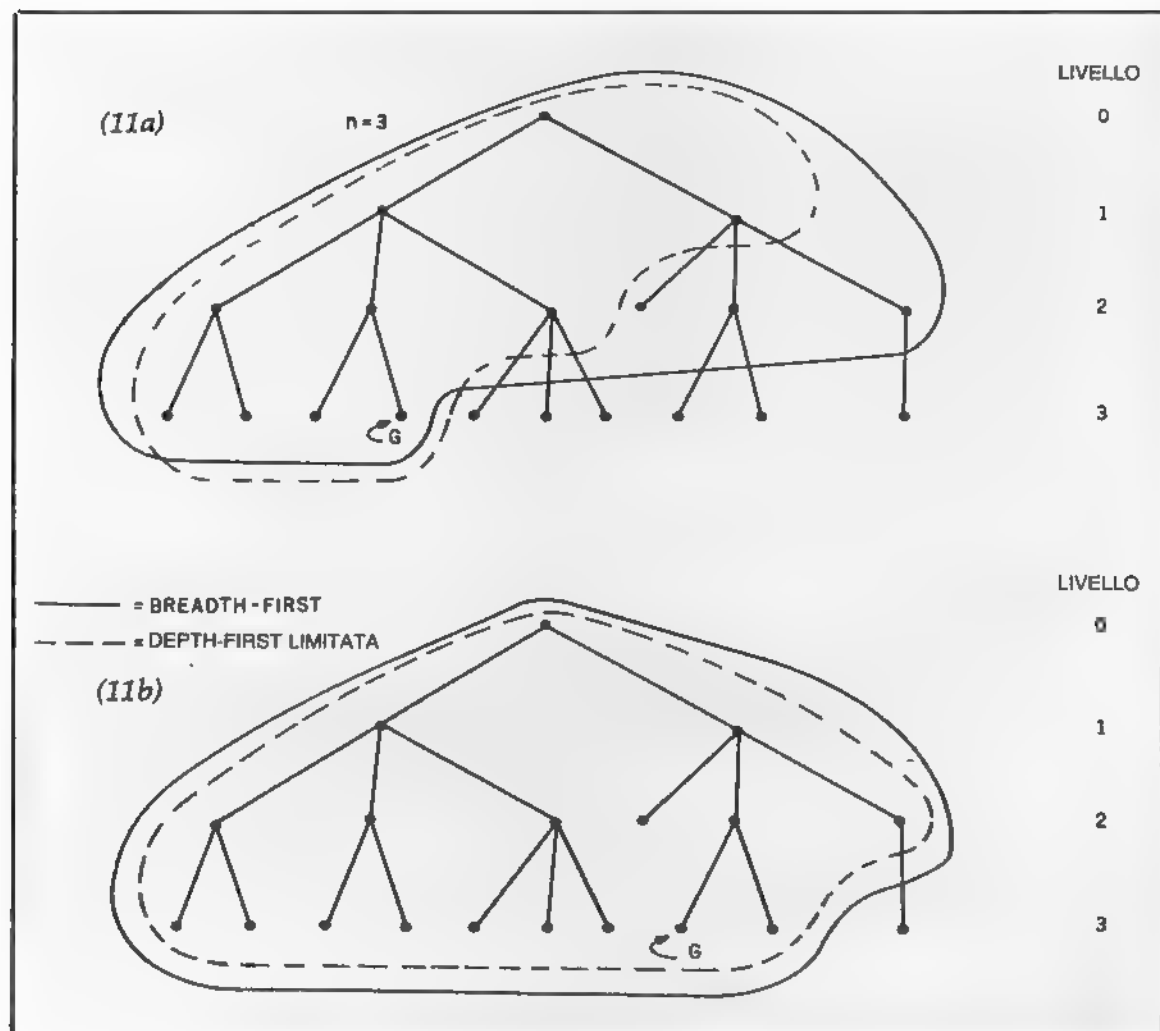



Fig. 11. *Un confronto di efficienza. La ricerca depth-first limitata produce sempre un numero di nodi minore o uguale della breadth-first quando il limite di ricerca è uguale al livello della meta. La figura 11a mostra un esempio dell'efficienza del metodo depth-first limitato rispetto al breadth-first. Nella figura 11b, i due tipi di ricerca generano lo stesso numero di nodi, con lo stesso albero ma una meta diversa. La tecnica breadth-first si estende sempre lungo tutto l'albero fino al livello n-1, mentre la versione limitata della depth-first può farne a meno se la meta è ben disposta.*

● Ho fatto una scoperta interessante: è sicuro che la ricerca *depth-first* limitata si estende con numero di nodi diversi o uguale della *breadth-first quando il limite di ricerca (n) è uguale al numero di passi (s) nella soluzione.* Una dimostrazione intuitiva di questo fatto si può dedurre dall'esempio in figura 11. Inoltre, si può dedurre che le due tecniche siano più o meno allo stesso grado di efficienza quando *n* è solo di poco maggiore di *s*.

● (Domanda 4) In una ricerca *depth-first* limitata al livello *n* per un problema risolvibile in *n* mosse, quali sono le due caratteristiche (per quanto riguarda la posizione della meta) che influenzano principalmente il tempo di soluzione? Qual è la più importante?

● Con la ricerca depth-first limitata a livello maggiore del numero di mosse nella soluzione del problema ($n>s$), il numero di nodi estesi cresce come la differenza tra i due (si controllino i risultati della tavola 3b).

● Quando il limite di ricerca è inferiore al numero di mosse della soluzione ($n<s$), si hanno due possibilità. Se s è multiplo di *n*, l'efficienza della ricerca è intermedia tra quelle dei metodi *breadth-first* e *depth-first* limitata a livello $n=s$ (come si vede nella figura 12). Tuttavia, se *s* non è multiplo di *n*, si hanno risultati misti. La ricerca *depth-first* si spinge a fondo parecchi livelli sotto quello della soluzione e può rivelarsi migliore della ricerca *breadth-first* o peggiore a seconda che il numero di nodi estesi superi il livello *s* ed a seconda del numero di nodi estesi dalla ricerca *breadth-first* che non sono estesi dalla *depth-first* limitata (vedi figura 12).

● (Domanda 5) Un caso speciale di ricerca depth-first limitata si ha quando il limite è uno ($n=1$). Come si chiama questa ricerca particolare?

Risposte

1. Se un certo nodo della soluzione è uno degli ultimi da estendere (diciamo in alto o a destra),

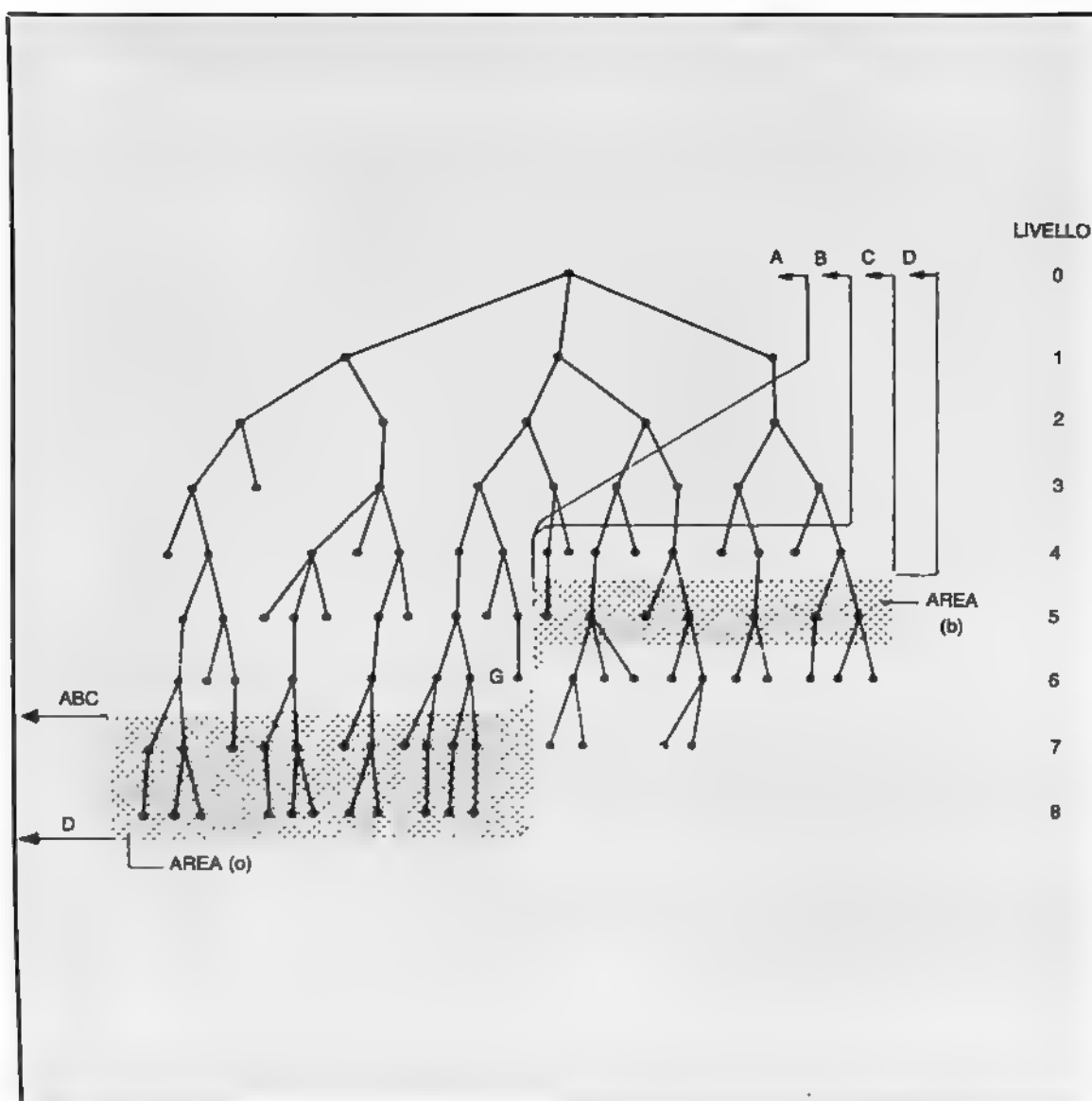



Fig. 12. *Confronto tra quattro metodi esaustivi di ricerca. Dato l'albero parziale con meta al livello sei, le quattro linee A, B, C e D separano i nodi estesi dalle seguenti tecniche:*
a) ricerca depth-first limitata con n=s=6
b) ricerca depth-first limitata con s multiplo pari di n (s=6,n=3)
c) ricerca breadth-first
d) ricerca depth-first limitata con s non multiplo pari di n (s=6,n=3).

allora la ricerca breadth-first si estenderà prima in altri modi improduttivi (tenete presente l'ordine: in basso, a sinistra, in alto, a destra).

È l'estensione di questi nodi che accresce il numero di nodi della lista CLOSED.

2. Sì, le mosse S vengono estese prima delle A, ma una analisi della situazione mostra che l'ordine con cui si estendono i nodi (basso, sinistra, alto, destra) è solo uno dei tanti fattori che influenzano il numero di nodi estesi dal metodo *breadth-first*. Consideriamo le figure 9 e 10. Se contate il numero di nodi terminali (che devono ancora essere estesi) e di nodi intermedi (già estesi), vedrete che esso cresce come quelli elencati nella tavola 2a. (Il nodo di partenza deve essere considerato chiuso; inoltre, per il momento, ignoriamo il nodo C della figura 9).

I due alberi sono identici come numero e distribuzione di nodi fino al livello due (nodi da 1 a 9). La ragione principale per cui lo schema (3,1) estende più nodi del (3,2) è che il primo deve estendere più nodi a livello due per trovare la meta. Perché? Il cammino che va dal nodo 3 (di entrambi gli alberi) alla meta nel problema (3,2) – il ramo S in figura 10 – si estende prima del corrispondente – il ramo A in figura 9.

In questo caso l'S nella soluzione dello schema (3,2) è più importante dei due S nella soluzione di (3,1), perché previene l'estensione di parecchi nodi "fertili" (il sei ed il sette nella figura 9) nel penultimo livello.

Tuttavia, non è il caso di generalizzare. I fattori che influenzano il numero di nodi generati sono interdipendenti al punto che è impossibile calcolare i loro rapporti di forza al di fuori del contesto di un esempio specifico.

3. Consideriamo la figura 9. L'intero albero (tranne il nodo C) è l'albero di soluzione dello schema (3,1). I nodi nell'area colorata (compreso C) sono l'albero di soluzione per (2,1). Risulta quindi chiaro il motivo per cui il nodo C viene generato solo da (2,1). Se il nodo B è di partenza, tutte quattro le direzioni generano successori validi, ma se B è stato generato da A, nodo iniziale del problema (3,1), il programma RICERCA ignora il nodo C, poiché la direzione di passaggio da B a C è opposta alla precedente, che trasforma A in B. Di nuovo, l'interazione di forze all'interno dell'algoritmo dà luogo a risultati inattesi ma corretti.

4. I due fattori importanti in questa situazione sono la "profondità" e la "distanza da sinistra" della meta. Poiché l'algoritmo depth-first limitato scandisce i nodi da sinistra a destra, la "distanza da sinistra" della meta è il fattore principale: un nodo che si trovi nel mezzo dell'albero ma più a sinistra di un altro verrà esteso prima. E ciò è vero anche per la ricerca depth-first pura.

5. Una ricerca depth-first limitata al livello 1 è equivalente alla ricerca *breadth-first*; l'algoritmo effettua una ricerca esaustiva su tutto il livello 1 dell'albero. La numerazione dei nodi secondo l'algoritmo depth-first limitato dimostrerà l'equivalenza dei due metodi.

Conclusioni

La seconda parte si occuperà di algoritmi euristici che stimano il "valore" di un dato nodo per arrivare sicuramente ad una soluzione senza effettuare una ricerca esaustiva.

Per ordinare il volume utilizzare l'apposito tagliando inserito in fondo alla rivista.

Z-80

Pag. 540 **L. 24.000** (Abb. L. 21.600)
Cod. 328D Formato 14,5 x 21

Questi due libri sono stati ideati come testi autonomi e completi per imparare la programmazione in linguaggio Assembler, usando lo Z80 o il 6502 (i microprocessori forse più diffusi).
Scorrevoli da leggere, non richiedono alcuna conoscenza di base, ne di elettronica generale né di programmazione.
Sono stati progettati, infatti, sotto forma di corso che, sistematicamente, passo dopo passo, porta il lettore dai concetti di base fino alle tecniche di programmazione avanzate, al fine di permettergli la realizzazione di programmi sempre più complessi.
L'esposizione progressiva, rigorosamente strutturata, comporta la risoluzione obbligatoria di esercizi attentamente graduati al fine di verificare che si sia veramente capito quanto presentato?
Ben si prestano, perciò, a chi si avvicina per la prima volta ai microprocessori e ne vuole conoscere e capire gli aspetti essenziali di programmazione. Per tutti coloro che già hanno programmato, invece, sarà una vera e propria miniera di informazioni sulle caratteristiche specifiche del microprocessore d'interesse, evidenziandone, nel contempo, vantaggi e svantaggi.

PROGRAMMAZIONE DELLO Z80
Rodnay Zaks

6502

Pag. 384
L. 22.000
(Abb. L. 19.800)
Cod. 503B
Formato 14,5 x 21

Programmazione del 6502
EDIZIONE ITALIANA
RODNAY ZAKS

La Potenza dei Microprocessori

SOMMARIO
Concetti Fondamentali; Organizzazione Hardware del Microprocessore; Tecniche Fondamentali di Programmazione; Set di Istruzioni; Tecniche di Indirizzamento; Tecniche di Input/Output; Dispositivi di Input/Output; Esempi Applicativi; Strutture dei Dati; Sviluppo del Programma; Conclusioni.

GRUPPO EDITORIALE JACKSON
Divisione Libri

Come tradurre i termini stranieri?

Lexikon

Scriveva Machiavelli: "...Le lingue non possono essere semplici, ma conviene che sieno miste con l'altre lingue. Ma quella lingua si chiama d'una patria, la quale convertisce i vocaboli che ha accettati nell'uso suo, ed è sì potente, che i vocaboli accettati non la disordinano, ma ella disordina loro, perché quello ch'ella reca da altri, lo tira a sé in modo che par suo".

In altre parole: nessun ostracismo alle parole straniere; ma la lingua, la nostra in questo caso, ha un carattere, uno stile, un'originalità da difendere: non solo rifiutando la parola straniera quando è del tutto inutile, ma applicando le regole della grammatica nel declinare i termini stranieri, quando questi non si possono o non si vogliono evitare.

Credo non esista un settore della nostra lingua dove queste riflessioni siano più appropriate, del "gergo" dell'informatica. La massa di termini angloamericani che ha investito la lingua italiana è andata via via crescendo negli ultimi dieci o venti anni. Come si può mettere un po' d'ordine in questa materia, chiarendo le idee agli addetti ai lavori e ai neofiti, obbligati ad usare questo gergo?

Vi sono tre semplici regole. Prima regola: tradurre *quando è possibile*. Attenzione, qui non si tratta di tradurre *a tutti i costi* trovando termini strampalati (come qualcuno fa, volendo tradurre *file* con *flusso* o *software* con *logicale*) o adattando in modo non sincero i termini stranieri all'italiano (*digitare*, *inputare*). Si tratta appunto di tradurre quando è possibile, quando la sensibilità linguistica dice che si può fare: *statement* si traduce con *istruzione*, *printer* con *stampante*, *random* con *casuale*.

Seconda regola. Se non è possibile tradurre, ma si ritiene che il termine straniero possa essere accettato nel linguaggio italiano (per lo meno nel linguaggio settoriale dell'informatica, come *software* e *hardware* per esempio, o *record*, *bit*, *byte*,...) bisogna ricordare che la grammatica prescrive di lasciare immutati al plurale i nomi che terminano in consonante: perciò come scriviamo e pronunciamo i *film*, gli *sport*,i *bar*, i *tram* (e non *films*, *sports*, *bars*, *trams*) dobbiamo scrivere e pronunciare i *file*, i *record*, i *bit* e i *byte*. Quante volte si vedono "un milione di bytes" e "due files"! Chi aggiunge questa "s" credendo di dimostrare di conoscere l'inglese, fa invece vedere di non conoscere l'italiano.

Ed infine la terza regola. Se non è possibile tradurre, e d'altro canto non si ritiene di accettare il termine straniero nel linguaggio italiano, bisognerà allora usare il termine come tale, cioè come straniero; e bisognerà scriverlo ovviamente nella forma plurale della sua lingua: le *utilities*, i *bugs*, i *jobs*. In tal caso andranno sottolineati in un testo manoscritto o messi in corsivo in un testo a stampa.

Queste sono le tre regole. Ma rileggendo ora la frase di Machiavelli, mi accorgo che una interpretazione restrittiva arriverebbe a concludere che egli consiglia di prendere, per esempio, *to bootstrap* e farlo diventare *bootstrappare*, *to shift* e farlo diventare *shiftare*. In questo senso Machiavelli è superato dall'uso attuale.

Ormai non sono più i tempi in cui Décartes si faceva diventare Cartesio, né quelli in cui si pretendeva di sostituire *gas* con *gasse*, *taxì* con *tassì*. Questa capacità di "disordinare", lodata dal Machiavelli, è per fortuna quasi del tutto scomparsa.

Cos'è successo a S. Pietro?

TEXAS INSTRUMENTS

Con l'Home Computer Texas Instruments potete conversare nei cinque principali linguaggi: BASIC, PASCAL, TI-LOGO, ASSEMBLER e INGLESE.

Se paragonate l'Home Computer TI-99/4A con i suoi concorrenti scoprirete che è un computer veramente eccezionale.

Tanto per cominciare, vi consente di usare in programmazione i più importanti linguaggi: una qualità che è difficile trovare in altri computer simili. Ma soprattutto ha una capacità RAM disponibile all'utente di ben 16 K byte espandibile a 48 K byte. Con l'aggiunta di un modulo «Solid State Software»® può raggiungere una capacità combinata RAM/ROM di 110 K byte.

L'Home Computer TI-99/4A si può collegare ad un normale apparecchio televisivo e può espandersi fino a diventare un sistema computerizzato completo con l'aggiunta di unità periferiche come ad esempio due normali registratori a cassetta, unità di controllo a distanza, memorie a disco, sintetizzatore della voce e stampante termica.

Grazie alla interfaccia opzionale RS 232 possono essere collegate altre unità periferiche quali modems di comunicazione, stampanti ad impatto e plotters.

Bisogna poi aggiungere la sua alta risoluzione grafica (256 x 192 punti), la capacità di operare con 32 caratteri su 24 linee in 16 colori, quella di emettere 3 tonalità in 5 ottave e di generare effetti sonori, quella di parlare grazie ad un sintetizzatore vocale e di conversare in BASIC, UCSD-PASCAL, TI-LOGO, ASSEMBLER: scoprirete che l'Home Computer TI-99/4A non può certo essere paragonato con i concorrenti. Soprattutto per quanto riguarda il prezzo: 598.000 lire IVA esclusa!

Se volete risolvere qualsiasi tipo di problema, potete usare la vasta gamma di moduli «Solid State Software»® Texas Instruments il cui uso è facilissimo.

Inoltre ci sono già 600 programmi software disponibili in tutto il mondo.

Dopotutto, è più che naturale aspettarsi alta tecnologia e prezzo accessibile da chi ha inventato il microprocessore, il circuito integrato e il microcomputer.

Vi aiutiamo a migliorare.

TEXAS INSTRUMENTS

Un assembler Basic per il PET/CBM

Due programmi (un editor e un assembler) per programmare in linguaggio macchina

di *R. Baker*

Questo semplice ma potente assembler a due passi per il PET/CBM Commodore genera direttamente file su disco di programmi caricabili. È scritto interamente in Basic, e dunque può girare su tutti i modelli. Tutti i comandidi disco sono spediti attraverso il canale di comandi-disco, e dunque può funzionare sia con DOS 1.0 (dischi 2040 e 3040) sia con DOS 2.x (dischi 4040 o 8050). Il programma richiede un drive per floppy disc Commodore e almeno 8 Kbyte di memoria. Per ottenere il massimo vantaggio dalle caratteristiche del programma è utile una stampante.

Il programma è stato scritto per schermo a 40 colonne e non è stato modificato per le nuove macchine a 80 colonne.

Sebbene questo assembler sia scritto in Basic, il tempo d'assemblaggio medio per ogni linea di codice sorgente è da uno a due secondi sul secondo passo. Ho tentato di migliorarne le prestazioni limitando le funzioni disponibili ed eseguendo alcuni controlli prelimi-

n	numero decimale esempio 12369
$n	numero esadecimale con prefisso $ esempio $A0F2
simbolo	valore simbolico, valore del simbolo specificato, esempio TAG2
*	il valore attuale del program counter
car	letterale ASCII, valore ASCII del primo carattere dopo l'apostrofo, esempio 'S

Tavola 1. Operandi delle istruzioni e delle direttive.

valore *o* +valore	valore positivo dell'operando
−valore	valore negativo in complemento a due
valore1 + valore2	somma i due valori
valore1 − valore2	sottrae i due valori
valore1/valore2	divisione dei due valori

I risultati sono valori a 16 bit.

Tavola 2.

Personal Software vi dà la possibilità di avere un dischetto con i due programmi di questo articolo già battuti, corretti e funzionanti.
In fondo all'articolo tutte le indicazioni.

```
LOC   OGGETTO    LINEA..........CODICE SORGENTE..............  DASM V3.0

                  1  *
                  2  * ESEMPI MODI DI INDIRIZZAMENTO
                  3  *
0025  6D 34 12    4           ADC    $1234         ASSOLUTO
0028  69 78       5           ADC#   $5678         IMMEDIATO
002A  65 01       6           ADCZ   1             PAGINA ZERO
002C  71 02       7           ADC@Y  2             INDICE INDIRETTO REG. Y
002E  61 03       8           ADC@X  3             INDICE INDIRETTO REG. X
0030  75 04       9           ADCZX  4             INDICE PAGINA ZERO REG. X
0032  B6 05      10           LDXZY  5             INDICE PAGINA ZERO REG. Y
0034  6C 06      11           JMP@   6             INDIRETTO
0036  D8         12           CLD                  IMPLICITO
0037  0A         13           ASLA                 ACCUMULATORE
0038  F0 01      14           BEQ    *+1           RELATIVO
003A  7D CD AB   15           ADCX   $ABCD         INDICE ASSOLUTO REG. X
003D  79 00 04   16           ADCY   1024          INDICE ASSOLUTO REG. Y
                 17  *
                 18  * OPERANDI E ESPRESSIONI
                 19  *
                 20  TAG      =      'A            LETTERALE ASCII
                 21  TAG2     =      'XYZ          LETTERALE ASCII; PRIMO C
                 22  ABC      =      1+$24         DEC+ESA
                 23  X2G      =      150/11        DIVISIONE
0040  F0 04      24           BEQ    *+4           SALTO RELATIVO
0042  D0 F8      25           BNE    *-8           
                 26  YYY      =      'B+128        LETTERALE ASCII+DEC
                 27  T2       =      -1            NUMERO NEGATIVO IN COMP.2
                 28  T3       =      T2-2          SOTTRAZIONE
                 29  *
                 30  * ERRORI ASSEMBLER
                 31  *
                 32           XXX                  COMANDO/OPERANDO
****  ERRATO....              ^ COMANDO
0044  00 00      33           ASL@X                INDIRIZZAMENTO
****  ERRATO....                ^ MODO DI INDIRIZZAMENTO
0046  F0 00      34           BEQ    *+200         SALTO ERRATO
****  ERRATO....                     ^ SALTO
0048  D0 00      35           BNE    *-200
****  ERRATO....                     ^ SALTO
                 36  TS       =      $XYZ          OPERANDO
****  ERRATO....                     ^ OPERANDO
                 37  TS       =      1000          SIMBOLO DUPLICATO
                 38  TS       =      123           SECONDA DUPLICAZIONE


                 SIMBOLO    ESA-VALORE DECIMALE

                 ABC           0025          37
                 T2            FFFF       65535
                 T3            FFFD       65533
                 TAG           0041          65
                 TAG2          0058          88
                 TS            03E8        1000
                 ******  DEFINIZIONE MULTIPLA
                 X2G           000D          13
                 YYY           00C2         194

                   6  ERRORI
```

```
10 REM ************* DASM EDITOR *************
20 REM *                                     *
30 REM *          DI ROBERT BACKER           *
40 REM *                                     *
50 REM ***************************************
60 REM
70 REM
80 PRINT CHR$(147);SPC(8);CHR$(18)"D A S M   E D I T O R"
90 CLR: U$=CHR$(17): FOR I=1 TO 40: U$=U$+CHR$(192): NEXT I: U$=U$+CHR$(145)
100 B$=" ": FOR I=1 TO 31: B$=B$+" ": NEXT I
110 O$(1)=CHR$(18)+" CREA "+CHR$(146): O$(2)=CHR$(18)+" EDIT "+CHR$(146)
115 O$(3)=CHR$(18)+"COPIA "+CHR$(146): O$(4)=CHR$(18)+"LEGGI "+CHR$(146)
120 O$(5)=CHR$(18)+"STAMPA"+CHR$(146): O$(6)=O$(5)
125 O$(7)=CHR$(192)+CHR$(192)
130 OPEN 15,8,15: CR$=CHR$(13): PRINT U$
140 PRINT" 0  -  "CHR$(18)" FINE ": PRINT
150 PRINT" 1  -  ";O$(1);O$(7);O$(7)" NUOVO FILE SORGENTE": PRINT
160 PRINT" 2  -  "; O$(2);O$(7);CHR$(174)
165 PRINT SPC(15);CHR$(171);CHR$(192)" VECCHIO FILE SORGENTE"
170 PRINT" 3  -  "; O$(3);O$(7);CHR$(179): PRINT SPC(15);CHR$(221)
180 PRINT" 4  -  "; O$(4);O$(7);CHR$(179): PRINT SPC(15);CHR$(221)
190 PRINT" 5  -  "; O$(5);O$(7);CHR$(189): PRINT
200 PRINT" 6  -  ";O$(6);O$(7);O$(7)" FILE LISTATO"
```

nari sui campi con il programma *editor* al momento della creazione del codice sorgente.

Il maggior vantaggio di questo assembler è che può assemblare programmi comunque grandi, poiché il file sorgente viene memorizzato su disco. Inoltre, poiché l'output è un file di programma caricabile, l'intervallo di indirizzi usati non è limitato, sempre che ogni programma sia un blocco contiguo.

Editor

Il programma *editor* (listato 1) prevede diverse funzioni per creare e correggere file sorgente che saranno di input al programma *assembler*. Ogni funzione vi permette di avere una copia stampata del file sorgente che viene scritto o letto. Questo *editor* effettua diversi controlli per far in modo che la successiva esecuzione del programma *assembler* sia il più possibile veloce. Inoltre, l'assembler si aspetta le linee sorgente di uno specifico formato nel file di input. L'editor permette anche di stampare i file salvati su disco.

Come si crea un file sorgente

La funzione CREA produce un file sorgente originale dai dati introdotti dalla tastiera. Ogni linea di codice sorgente consiste di un numero di campi, variabile da uno a quattro: etichetta, comando, operando e commento. I campi etichetta, operando e commento sono opzionali, ma il campo comando deve essere sempre presente affinché la linea possa essere accettata dall'*editor*.

L'unica eccezione a questa regola è la linea di tutto commento che comincia con un asterisco (*). L'*editor* non permette l'introduzione di più di un certo numero di caratteri per ogni campo, ed i caratteri non permessi vengono ignorati. Non si possono usare la virgola (,), i due punti (:) e le virgolette (") in nessun campo perché si tratta di caratteri che il PET utilizza in modo speciale.

```
210 PRINT U$: PRINT "SCEGLI LA FUNZIONE CHE DESIDERI:";
220 GOSUB 2610: F=V: IF R$="0" THEN PRINT CHR$(147): END
230 IF F<1 OR F>6 THEN 220
240 GOSUB 2250: IF F>4 THEN GOSUB 2310
250 IF F>3 THEN 270
260 PRINT"VUOI UNA COPIA STAMPATA";: GOSUB 2630: IF A=83 THEN GOSUB 2310
270 ON F GOTO 310,1540,1810,2080,2080,1970
280 REM ===========================
290 REM CREO UN NUOVO FILE SORGENTE
300 REM ===========================
310 GOSUB 2320: FL$=FL$+".SRC"
320 OPEN 2,8,2,DR$+":"+FL$+",S,W": GOSUB 2440: IF EN THEN 2210
330 GOSUB 2250
340 N=1
350 L$="": C$=L$: O$=L$: T$=O$: GOSUB 2470
360 REM ===============
370 REM CAMPO ETICHETTA
380 REM ===============
390 PRINT: PRINT CHR$(145); TAB(7);: C=1
400 GOSUB 2660: IF C=1 AND A=42 THEN 1090
410 IF A=32 THEN 600
420 S$=L$: IF (A>64) AND (A<91) THEN 550
430 IF(C>1) AND (A>47) AND (A<91) THEN 550
440 IF A=160 OR B=17 THEN 390
450 IF B=13 OR B=19 THEN 1370
460 IF A=157 THEN GOSUB 2710
470 IF A=29 THEN GOSUB 2690
480 T=6: IF A=20 THEN GOSUB 2730: GOTO 560
490 IF (A<>148) OR (LEN(L$)>5) THEN 400
500 GOSUB 2770
510 GOSUB 2660: IF A=141 OR B=19 THEN 1370
520 IF (A>64) AND (A<91) THEN 540
530 IF (C=1) OR (A<48) OR (A>59) THEN 510
540 GOSUB 2820: GOTO 560
550 IF C<7 THEN GOSUB 2540
560 L$=S$: GOTO 400
570 REM =============
580 REM CAMPO COMANDO
590 REM =============
600 PRINT TAB(15);: C=1
610 GOSUB 2660: IF A=32 THEN 750
620 S$=C$: IF (A>32) AND (A<96) AND (C<6) THEN GOSUB 2540: GOTO 710
630 IF (B=17) OR (C=1 AND A=160) THEN 390
640 IF A=160 THEN PRINT: PRINT CHR$(145);: GOTO 110
650 IF B=13 OR B=19 THEN 1370
660 IF A=157 THEN GOSUB 2710
670 IF A=29 THEN GOSUB 2690
680 T=14: IF A=20 THEN GOSUB 2730: GOTO 710
690 IF (A<>148) OR (LEN(C$)>4) THEN 610
700 GOSUB 2790: IF A=141 OR B=19 THEN 1370
710 C$=S$: GOTO 610
720 REM ==============
730 REM CAMPO OPERANDO
740 REM ==============
750 PRINT TAB(23);: C=1
760 GOSUB 2660: IF A=32 THEN PRINT: PRINT: GOTO 910
770 S$=O$: IF (A>32) AND (A<96) AND (C<14) THEN GOSUB 2540: GOTO 870
780 IF C=1 AND A=160 THEN PRINT: PRINT CHR$(145);: GOTO 750
790 IF A=160 THEN PRINT: PRINT CHR$(145);: GOTO 750
800 IF B=17 THEN 390
810 IF B=13 OR B=19 THEN 1370
820 IF A=157 THEN GOSUB 2710
830 IF A=29 THEN GOSUB 2690
840 T=22: IF A=20 THEN GOSUB 2730: GOTO 870
850 IF (A<>148) OR (LEN(O$)>12) THEN 760
860 GOSUB 2790: IF A=141 OR B=19 THEN 1370
870 O$=S$: GOTO 760
880 REM ==============
890 REM CAMPO COMMENTO
900 REM ==============
910 PRINT TAB(7);: C=1
920 GOSUB 2660: IF C<>1 OR A<>160 GOTO 930
925 PRINT: PRINT CHR$(145);CHR$(145);CHR$(145);: GOTO 750
930 IF A=160 THEN PRINT: PRINT CHR$(145);: GOTO 910
940 S$=T$: IF (B>31) AND (B<96) AND (C<26) THEN GOSUB 2540: GOTO 1050
950 IF B=17 THEN PRINT CHR$(145);CHR$(145);: GOTO 390
960 IF B=13 OR B=19 THEN 1380
970 IF A=157 THEN GOSUB 2710
980 IF A=29 THEN GOSUB 2690
990 T=6: IF A=20 THEN GOSUB 2730: GOTO 1050
1000 IF (A< >148) OR (LEN(T$)>24) THEN 920
1010 GOSUB 2770
1020 GOSUB 2660: IF A=141 OR B=19 THEN 1380
1030 IF (B<42) OR (B>95) THEN 1020
1040 GOSUB 2820
1050 T$=S$: GOTO 920
1060 REM ====================
1070 REM COMMENTO A PIENA RIGA
1080 REM ====================
1090 IF L$+C$+O$+T$<>"" THEN 400
1100 L$=R$: PRINT R$;
1110 PRINT TAB(9);: C=1
1120 GOSUB 2660: S$=T$: IF ((B>32) AND (B<96)) OR A=32 THEN 1370
1130 IF B=13 OR B=19 THEN 1370
1140 IF A<>157 THEN 1170
```

L'etichetta deve iniziare con una lettera e i rimanenti caratteri possono essere lettere o cifre. La sua lunghezza non può superare i sei caratteri. Il comando è o una istruzione mnemonica con l'appropriato suffisso di indirizzo o una direttiva per l'assemblatore. La lunghezza massima di questo campo è di cinque caratteri. Il campo operando contiene l'operando dell'istruzione (se necessario) e non deve essere superiore a 13 caratteri. Si veda la descrizione dell'assembler per maggiori informazioni sul formato e la sintassi dei campi comando e operando.

Il campo commento vi permette di inserire un commento di 25 caratteri in ogni linea. I caratteri possono essere alfanumerici o grafici, con le esclusioni sopra citate. È anche possibile introdurre un'intera linea di commento lunga fino a 50 caratteri invece dei quattro campi, battendo un asterisco come primo carattere della linea.

Ogni linea sorgente viene presentata sul video come due linee, con una linea vuota in mezzo per un puntatore. Come indicazione per l'input viene indicato il numero di linea corrente. I campi etichetta, comando e operando sono sulla linea superiore, ed il commento sulla linea inferiore. Se la linea è di tutto commento, occuperà le due linee visualizzate. Una freccia indicherà la posizione corrente sulla linea in ogni momento in cui può essere accettato l'input. Se scegliete l'opzione di stampa, ogni linea verrà stampata nel momento in cui viene scritta sul file di output.

Una volta scelta la funzione di creazione, i seguenti tasti hanno un effetto speciale:

- la barra dello spazio sposta il puntatore sul primo carattere del prossimo campo; uno spazio battuto in un campo commento verrà introdotto come spazio e non avrà altri effetti;
- lo shift assieme alla barra dello spazio riporta il puntatore sul primo carattere del campo attuale; se il puntatore è già sul primo carattere, passerà sul primo carattere del campo precedente;

• il tasto RETURN introduce la linea corrente se valida (cioè se è presente un comando o se è una riga di tutto commento).

• lo shift assieme al tasto RETURN scarta la linea corrente, elimina il numero di linea e permette di ripartire con la stessa linea;

• il tasto CLR/HOME termina il file sorgente scartando la linea corrente se è stata battuta ma non introdotta con il tasto RETURN;

• il tasto di cursore destra/sinistra sposta il puntatore a sinistra e a destra entro i limiti del campo corrente: il puntatore non può essere spostato oltre i caratteri esistenti nel campo o il prossimo spazio dopo l'ultimo carattere;

• il tasto INS permette di inserire un carattere prima del carattere puntato; dopo aver premuto il tasto INS deve essere introdotto un carattere, oppure shift e RETURN per scartare la linea; non si può usare INS per inserire più del massimo di caratteri di un campo;

• battendo il tasto DEL si cancella il carattere attualmente puntato.

Si noti che i tasti cursore su/giu e il tasto RVS sono ignorati.

Editing di un file sorgente

L'editor vi permette di correggere un file sorgente copiandolo da un nuovo file di output. L'editor terrà una copia del file sorgente originale, nel caso ci siano errori di sistema.

Se scegliete l'opzione di stampa, ogni linea scritta nel nuovo file verrà anche stampata. Quando la prima linea del file sorgente originale verrà visualizzata, scegliete il modo di edit desiderato tra quelli disponibili. Alla fine del file di input originale, potete aggiungere linee alla fine del file.

Facendo l'editing del file riga per riga, ogni linea viene visualizzata una linea alla volta e può essere editata come si desidera. Tutte le funzioni di controllo possono essere usate come descritto per la funzione di creazione del file con le seguenti differenze:

• la combinazione shift/RETURN

```
1150 IF C<>32 GOTO 1170
1160 PRINT: PRINT CHR$(145);CHR$(145);CHR$(145);CHR$(145): PRINT TAB(32);
1165 C=C-1: GOTO 1350
1170 IF A=29 AND C<LEN(T$)+1 THEN GOSUB 2700: GOTO 1350
1180 IF A< >160 AND B< >17 THEN 1210
1190 PRINT: PRINT CHR$(145);: IF C>30 THEN PRINT CHR$(145);CHR$(145);
1200 GOTO 1110
1210 IF (A< >148) OR (LEN(T$)>49) THEN 1280
1220 IFC>31THENPRINT" ";MID$(T$,C):PRINT"";TAB(70);" ":PRINTTAB(9);MID$(T$,31)
1230 PRINT" ";MID$(T$,C,31-C);TAB(70);" ":PRINT TAB(9);MID$(T$,31)
1240 PRINT CHR$(145);CHR$(145);CHR$(145);TAB(8+C)
1250 GOSUB 2660: IF A=141 OR B=19 THEN 1370
1260 IF (B<32) OR (B>95) THEN 1250
1270 S$=T$: GOSUB 2750: W$=MID$(T$,C): PRINT CHR$(145);TAB(C-23);: GOTO 1320
1280 IF (A< >20) OR (C>LEN(T$)) THEN 1120
1290 IF C>31 THEN PRINT MID$(T$,C+1);" ": PRINTCHR$(145);TAB(C-23);: GOTO 1320
1300 PRINT MID$(T$,C+1,32-C);" ";TAB(70);" ": PRINT TAB(9);MID$(T$,33);" "
1310 PRINT CHR$(145);CHR$(145);CHR$(145): SPC(8+C);
1320 S$=T$: R$=MID$(T$,C+1): GOSUB 2750: T$=S$: GOTO 1120
1330 IF C>50 THEN 1120
1340 GOSUB 2540
1350 IF C=32 THEN PRINT: PRINT SPC(9);
1360 T$=S$: GOTO 1120
1370 IF C<32 THEN PRINT: PRINT
1380 PRINT: PRINT: IF A=141 OR B=19 THEN 1410
1390 IF L$="*" OR C$< >"" THEN GOSUB 2380: GOSUB 2420: GOTO 1450
1400 PRINT CHR$(145);CHR$(145);CHR$(145);CHR$(145);: GOTO 390
1410 PRINT CHR$(145);CHR$(145);CHR$(145)CHR$(145)">>>>"CHR$(17);CHR$(17);
1415 PRINT CHR$(17): IF B< >19 THEN 1450
1420 IF M THEN 1610
1430 C$=".END": L$="": O$=L$: T$=O$: GOSUB 2430: IF M=0 THEN GOSUB 2680
1440 GOTO 2210
1450 IF M<>1 THEN 1490
1460 GOSUB 2560: IF E THEN 1770
1470 GOSUB 2500: PRINT CHR$(145);CHR$(145);CHR$(145);CHR$(145);
1475 IF L$="*" THEN 1110
1480 GOTO 390
1490 IF A<>141 THEN N=N+1
1500 GOTO 350
1510 REM ============
1520 REM EDITO IL FILE
1530 REM ============
1540 GOSUB 2320: E=0
1550 PRINT#15,"S"+DR$+":"+FL$+".BAK": INPUT#15.EN
1560 PRINT#15,"C"+DR$+":"+FL$+".BAK="+DR$+":"+FL$+".SRC"
1570 GOSUB 2440: IF EN THEN 2210
1575 PRINT#15,"S"+DR$+":"+FL$+".SRC": INPUT#15,EN
1580 OPEN 1,8,2,DR$+":"+FL$+".BAK,S,R": GOSUB 2440: IF EN THEN 2210
1590 OPEN 2,8,3,DR$+":"+FL$+".SRC,S,W": GOSUB 2440: IF EN THEN 2210
1600 GOSUB 2560: IF E THEN 1770
1610 M=0:GOSUB2250: GOSUB2490: PRINT U$:PRINT" 1  EDITO RIGA PER RIGA":PRINT
1620 PRINT"  2 - INSERISCO RIGHE PRIMA DI QUESTA": PRINT
1630 PRINT"  3 - COPIO DA QUESTA ALLA RIGA...": PRINT
1640 PRINT"  4 - CANCELLO DA QUESTA ALLA RIGA...": PRINT
1650 PRINT"CHE MODO DI EDITING DESIDERI?";
1660 GOSUB2610: M=V: IF V<1 OR V>4 THEN 1660
1670 GOSUB 2250: ON M GOTO 1470,340
1680 GOSUB 2500: PRINT U$: IF M=3 THEN PRINT"COPIO ";
1690 IF M=4 THEN PRINT "CANCELLO ";
1700 PRINT"DA QUESTA RIGA ALLA...";CHR$(17): PRINT"(FINE=FINE FILE)": PRINT
1710 PRINT"RIGA ";N:PRINT CHR$(145);SPC(4);
1720 INPUT R$: L=VAL(R$): IF R$="FINE"THEN L=1E5
1730 GOSUB 2250: IF L<N THEN 1680
1740 IF M=3 THEN PRINT "STO COPI";
1750 IF M=4 THEN PRINT "STO CANCELL";
1760 PRINT"ANDO LE RIGHE...": PRINT: PRINT: GOTO 1880
1770 GOSUB 2250: PRINT"FINE DEL FILE DI INPUT ORIGINALE": GOSUB 2680
1780 PRINT"VUOI AGGIUNGERE ALTRE RIGHE";
1790 GOSUB 2630: M=0: IF A=78 THEN M=2: GOTO 1430
1800 N=Z+1: GOSUB 2250:GOTO 350
1810 PRINT: PRINT"NOME FILE OUTPUT  ";CHR$(162);CHR$(157);CHR$(157);CHR$(157);
1815 INPUT FL$: IF FL$=CHR$(162) THEN 184
1820 GOSUB 2350:OPEN 2,8,3,DR$+":"+FL$+".SRC.S.W":GOSUB 2440:IF EN THEN 2210
1830 GOSUB 2250: E=0: N=0
1840 PRINT: PRINT"NOME FILE INPUT   ";CHR$(162);CHR$(157);CHR$(157);CHR$(157);
1845 INPUT FL$: IF FL$=CHR$(162) THEN 1840
1850 GOSUB 2350:OPEN 1,8,2,DR$+":"+FL$+".SRC.S.R":GOSUB 2440: IF EN THEN 2210
1860 GOSUB 2250: PRINT"STO COPIANDO IL FILE...": PRINT: PRINT
1870 GOSUB 2560: IF E THEN 1910
1880 IF M<4 THEN GOSUB 2380: GOSUB 2420
1890 IF M AND (N=L) THEN 1600
1900 GOTO 1870
1910 IF M THEN 1770
1920 CLOSE 1: GOSUB 2440: IF EN THEN 2210
1930 GOSUB 2680: PRINT"COPIO UN ALTRO FILE";
1940 GOSUB 2630: IF A=78 THEN GOSUB 2420: GOTO 2210
1950 GOTO 1830
1960 REM ==================
1970 REM STAMPO IL FILE LISTATO
1980 REM ==================
1990 GOSUB 2320: FL$=FL$+".LST"
2000 OPEN 1,8,4,DR$+":"+FL$+",S,R": GOSUB 2440: IF EN THEN 2210
2010 GOSUB 2250: PRINT: PRINT"STO STAMPANDO IL FILE.....": PRINT
2020 INPUT#1,L$: GOSUB 2440: IF EN>0 OR L$=".END" THEN 2210
2030 IF LEFT$(L$,1)="." THEN L$=" "+MID$(L$,2)
```

```
2040 PRINT#204,L$: GOTO 2020
2050 REM ============
2060 REM LEGGO IL FILE
2070 REM ============
2080 GOSUB 2320: FL$=FL$+".SRC"
2090 E=0: OPEN 1,8,2,DR$+":"+FL$+".S,R":GOSUB2440:IF EN THEN 2210
2100 GOSUB 2250: IF F=5 THEN PRINT: PRINT"STO STAMPANDO IL FILE...": PRINT
2110 GOSUB 2570: IF E THEN 2170
2120 IF F=4 THEN GOSUB 2500
2130 GOSUB 2380: GET R$: IF R$="" THEN 2110
2140 IF (ASC(R$)AND 127)=19 THEN PRINT U$: GOTO 2210
2150 GOSUB 2600: IF F=5 THEN 2100
2160 GOTO 2110
2170 IF F=4 THEN GOSUB 2600
2180 REM ===========================
2190 REM CHIUDO TUTTI I FILE E RIPARTO
2200 REM ===========================
2210 CLOSE 1: CLOSE 2: CLOSE 204: CLOSE 15: GOTO 80
2220 REM =========
2230 REM SUBROUTINE
2240 REM =========
2250 PRINT CHR$(147);O$(F);: IF M=0 THEN PRINT: GOTO 2300
2260 PRINT"  -  ";: IF M=1 THEN PRINT"RIGA PER RIGA"
2270 IF M=2 THEN PRINT "INSERIMENTO"
2280 IF M=3 THEN PRINT "COPIA"
2290 IF M=4 THEN PRINT "CANCELLAZIONE"
2300 PRINT U$: RETURN
2310 P=1: OPEN 204,4: RETURN
2320 PRINT: PRINT"NOME DEL FILE  "CHR$(162);CHR$(157);CHR$(157);CHR$(157);
2330 INPUT FL$: IF FL$=CHR$(162) THEN 2320
2340 FL$=LEFT$(FL$,12)
2350 PRINT: PRINT"DRIVE 0 O 1: ";
2360 GET DR$: IF DR$<>"0"AND DR$<>"1" THEN 2360
2370 PRINT DR$: RETURN
2380 IF P=0 THEN RETURN
2390 PRINT# 204,SPC(6-LEN(STR$(N)));N;SPC(2);
2400 IF L$="*" THEN PRINT# 204,"* ";T$: RETURN
2410 PRINT#204,L$;SPC(8-LEN(L$));C$;SPC(7-LEN(C$));O$;SPC(15-LEN(O$));T$
2415 RETURN
2420 Z=Z+1
2430 PRINT#2,L$;",";C$;",";O$;",";T$;CR$;
2440 INPUT#15,EN$,EM$,ET$,ES$:EN=VAL(EN$): IF EN=0 THEN RETURN
2450 GOSUB 2250: PRINT CHR$(18)"ERRORE DI DISCO":CHR$(17)
2455 PRINT EN$;" ";EM$,ET$;" ";ES$
2460 PRINT: PRINT: GOTO 2600
2470 V=4: IF M=2 THEN PRINT CHR$(18);CHR$(166);:V=3
2480 PRINT RIGHT$("    "+STR$(N),V);CHR$(146);" ";: RETURN
2490 L$=L1$: C$=C1$: O$=O1$: T$=T1$: N=N1
2500 GOSUB 2470: PRINTTAB(7);L$;
2510 IF L$<>"*" THEN 2530
2520 PRINT" ";LEFT$(T$,31);TAB(50);"": PRINTTAB(9);MID$(T$,32): PRINT: RETURN
2530 PRINTTAB(15);C$;TAB(23);O$:PRINT :PRINTTAB(7);T$: PRINT: RETURN
2540 IF C>LEN(S$) THEN S$=S$+R$: GOTO 2700
2550 W$=MID$(S$,C+1): GOSUB 2750: S$=S$+W$: GOTO 2700
2560 GOSUB 2570: N1=N: L1$=L$: C1$=C$: O1$=O$: T1$=T$: RETURN
2570 INPUT#1,L$,C$,O$,T$: GOSUB 2440
2580 IF C$=".END" THEN E=1: RETURN
2590 N=N+1: RETURN
2600 PRINT CHR$(18)"       BATTI UN TASTO PER CONTINUARE     "
2610 GET R$: IF R$="" THEN 2610
2620 V=VAL(R$): A=ASC(R$): B=(A AND 127): RETURN
2630 PRINT" (S/N): ";
2640 GOSUB 2610: IF A<>83 AND A<>78 THEN 2640
2650 PRINT R$: PRINTU$: RETURN
2660 PRINT CHR$(17)"^"CHR$(157);:GOSUB 2610: PRINT" "CHR$(157);CHR$(145);
2665 IF A=34 OR A=44 OR A=58 THEN 2660
2670 RETURN
2680 PRINT: PRINT"NUOVO FILE=";Z;"LINEE": PRINT U$: RETURN
2690 IF C>LEN(S$) THEN RETURN
2700 C=C+1: PRINT R$;: RETURN
2710 IF C>1 THEN C=C-1: PRINT R$;
2720 RETURN
2730 IF C>LEN(S$) THEN RETURN
2740 R$=MID$(S$,C+1): PRINT R$;" ": PRINT CHR$(145);TAB(T+C);
2750 IF C>1 THEN S$=LEFT$(S$,C-1)+R$: RETURN
2760 S$=R$: RETURN
2770 W$=MID$(S$,C): PRINT" ";W$;
2780 PRINT: PRINT CHR$(145);TAB(T+C);: RETURN
2790 GOSUB 2770
2800 GOSUB 2660: IF A=141 OR B=19 THEN RETURN
2810 IF (A<33) OR (A>95) THEN 2800
2820 GOSUB 2750: S$=S$+W$: GOTO 2700
```

cancella la linea corrente e visualizza la prossima linea;

- il tasto CLR/HOME visualizza la linea corrente e ritorna a selezionare un altro modo di edit a quel punto
- l'opzione "inserire una linea prima di quella attuale" permette di aggiungere nuove linee di codice sorgente prima della linea corrente, esattamente come nella creazione di un nuovo file. Battendo CLR/HOME viene visualizzata la linea corrente e potete selezionare un altro modo di edit;
- l'opzione "copia da una linea a un'altra" copia alcune linee nel nuovo file senza cambiamenti mentre ogni linea viene visualizzata ed eventualmente stampata. Se non viene introdotto alcun numero e si preme RETURN, solo la linea corrente verrà copiata. Battendo la parola FINE, il file di input verrà copiato fino alla fine;
- l'opzione "cancella da una linea a un'altra" è simile all'opzione di copia eccetto che cancella le linee e visualizza quelle rimanenti.

Copia di file sorgente

Questa funzione dell'editor copia un file sorgente già esistente. Ogni linea viene visualizzata ed eventualmente stampata. File multipli si possono copiare e concatenare in un nuovo file sorgente. Alla fine di ogni file copiato, il numero totale di linee nel nuovo file viene visualizzato.

Leggere e stampare un file sorgente

Questa funzione legge un file sorgente esistente e visualizza e/o stampa ogni linea al momento della lettura. Battendo un tasto eccetto CLR/HOME o RUN/STOP la funzione alternativamente si ferma e riparte permettendovi di esaminare le varie sezioni. Battendo CLR/HOME la funzione termina, il file di input viene chiuso e si ritorna al menu principale. Se non state usando una stampante Commodore probabilmente dovete modificare la routine di stampa affin-

```
10 REM ************* DASM ASSEMBLER *************
20 REM *                                         *
30 REM *              DI ROBERT BAKER            *
40 REM *                                         *
50 REM *******************************************
80 PRINT CHR$(147);SPC(6);CHR$(18)"D A S M   A S S E M B L E R": PRINT: PRINT
90 CLR: CR$=CHR$(13)
100 SZ=100: DIM S$(SZ),V(SZ)
110 PRINT: PRINT"NOME DEL FILE  "CHR$(162)CHR$(157)CHR$(157)CHR$(157);
111 INPUT FL$
120 IF FL$=CHR$(162) THEN 110
130 PRINT: PRINT"DRIVE 0 O 1: ";
140 GET DR$: IF DR$<>"0" AND DR$<>"1" THEN 140
150 PRINT DR$: FL$=LEFT$(FL$,12): OPEN 15,8,15
160 PRINT#15,"S"+":"+FL$: INPUT#15,EN
170 GOSUB 1490
180 OPEN 2,8,3,DR$+":"+FL$+",P,W": GOSUB 1510
190 PRINT: PRINT"DEVO LISTARE SUL DISCO (D)": PRINT"O SULLA STAMPANTE (S): ";
200 GET PR$: IF PR$<>"D" AND PR$<>"S" THEN 200
210 PRINT PR$: IF PR$<>"S" GOTO 220
215 OPEN 204,4: LE$=CHR$(13)+CHR$(10): LS$=" ": GOTO 270
220 PRINT#15,"S"+DR$+":"+FL$+".LST": INPUT#15,EN
230 OPEN 204,8,4,DR$+":"+FL$+".LST,S,W": GOSUB 1510: LE$=CHR$(13): LS$="."
240 REM ===========================================
250 REM PRIMO PASSAGGIO, GENERO LA TAVOLA DEI SIMBOLI
260 REM ===========================================
270 PRINTCHR$(147)"PRIMO PASSO - DEFINIZIONE DEI SIMBOLI": PRINT: PRINT
280 GOSUB 1500: IF L$<"A" THEN 350
290 PRINT L$,
300 IF S<>SZ AND FRE(0)>=100 GOTO 310
305 PRINT CHR$(18)"OVERFLOW DELLA TAVOLA DEI SIMBOLI": GOTO 440
310 S=S+1: S$(S)=L$: IF C$ "=" THEN 340
320 GOSUB 1160: V(S)=N: IF N 0 THEN S=S-1
330 GOTO 280
340 V(S)=P
350 IF C$>"@" THEN GOSUB 1100: GOTO 280
360 IF C$<>"*=" THEN 410
370 IF P>0 THEN 280
380 GOSUB 1160: IF N>-1 THEN P=N
390 IF L$<>"" THEN V(S)=P
400 GOTO 280
410 IF C$=".BY" THEN P=P+1
420 IF C$=".ADR" THEN P=P+2
430 IF C$< ".END" THEN 280
440 CLOSE 1: GOSUB 1510: GOSUB 1490
450 REM ===========================================
460 REM SECONDO PASSAGGIO.GENERO L'OGGETTO
470 REM ===========================================
480 PRINTCHR$(147)"SECONDO PASSO - LISTA/OGGETTO"
490 L=0: PS=0: EN=0: M$="....."
500 PRINT#204,"LOC   OGGETTO    LINEA";M$;M$;
510 PRINT#204,"CODICE SORGENTE";M$;M$;M$;" <DASM V3.0>";LE$;
520 PRINT#204,LS$;"  ";LE$;
530 E=0: B$="": GOSUB 1500: L=L+1: N$=STR$(L): IF L$="" THEN 560
540 IF L$="*" THEN PRINT#204,LS$;SPC(19-LEN(N$));N$;"  * ";T$;LE$;: GOTO 530
550 IF C$="=" THEN 890
560 IF C$>"@" THEN 670
570 IF C$<>"*=" THEN 630
580 IF PF>0 THEN E=E+4: GOTO 900
590 GOSUB 1160: IF N<0 THEN E=E+2: GOTO 900
600 P=N: PS=P
610 GOSUB 620: GOTO 770
620 P1=INT(PS/256): PRINT#2,CHR$(PS-(P1*256)); CHR$(P1);: PF=1: RETURN
630 PS=P: PB=3: IF C$=".BY" THEN P=P+1: GOTO 770: REM  - 1-BYTE COSTANTE
640 IF C$=".ADR" THEN P=P+2: GOTO 730: REM <== 2-BYTE ADR COSTANTE
650 IF C$=".END" THEN 940
660 E=E+4: GOTO 900
670 PB=0: GOSUB 1100: I$=MID$(I$,5): IF PB=0 THEN E=E+4: GOTO 900
680 A$=LEFT$(A$+"  ",2)
690 IF LEFT$(I$,2)=A$ THEN 720
700 I$=MID$(I$,6): IF I$ "" THEN 690
710 E=E+8: X1=PS: GOTO 1450
720 N=VAL(MID$(I$,3,3)): GOSUB 1420: GOSUB 1460: B$=H2$+" ":IF PB=1 THEN 770
730 GOSUB 1160: IF N 0 THEN E=E+.: GOTO 1440
740 IF PB=2 THEN N=N-P: IF N 127 OR N -128 THEN E E+16: GOTO 1440
750 N1=INT(N/256): N2=N-(N1*256): GOSUB 1460: N=N2: GOSUB 1420: B$=B$+H2$+" "
760 IF P-PS=3 OR C$=".ADR" THEN N=N1: GOSUB 1420: B$ B$+H1$
770 N=PS: GOSUB 1460: PRINT#204,H$;
780 PRINT#204,"  ";B$;SPC(14-LEN(B$)-LEN(N$));N$;"  ";L$;SPC(8-LEN(L$));C$;
790 PRINT#204,SPC(7-LEN(C$));O$;SPC(15-LEN(O$));T$;LE$;: IF E=0 THEN 530
800 REM ===========================================
810 REM FLAG ERRORS IN LISTING
820 REM ===========================================
830 PRINT#204,"****  ERRATO.... ;SPC(14);
840 IF (E AND 2) THEN PRINT#204,SPC(7);"  OPERANDO"
850 IF (E AND 4) THEN PRINT#204,"  COMANDO
860 IF (E AND 8) THEN PRINT#204,SPC(7);"  MODO DI INDIRIZZAMENTO"
870 IF (E AND 16) THEN PRINT#204,SPC(7);"  SALTO"
880 EN=EN+1: GOTO 530
890 GOSUB 1160: IF N 0 THEN E=E+2
900 PRINT#204,LS$;SPC(7);: GOTO 780
910 REM ===========================================
915 REM STAMPO LA TAVOLA DEI SIMBOLI
920 REM  % IL SOMMARIO DEGLI ERRORI
930 REM ===========================================
```

ché funzioni adeguatamente. Se non usate una stampante, rispondete N (no) alla domanda sulla stampante all'inizio dell'editor. In questo modo, tutta la stampa viene automaticamente eliminata.

Stampa di file di listati

Questa funzione semplicemente stampa il listato salvato su disco prendendolo da disco e stampandolo su stampante.

Assembler

Il programma assembler (listato 2) è un semplice assembler a due passi che legge il file sorgente generato dal programma editor e genera un file programma standard su disco. È anche prevista una stampa della lista di assemblaggio completa con una tavola dei simboli che può opzionalmente venir salvata su disco.

Durante il primo passo, l'assembler costruisce la tavola dei simboli definendo ogni etichetta e calcolandone il valore. I simboli sono visualizzati man mano che sono definiti per darvi la sensazione di come procede il primo passo. La tavola dei simboli è attualmente limitata a 100 simboli. Questo limite può essere cambiato modificando il valore di SZ nella linea 100. Se viene raggiunto il limite della tavola, oppure la memoria libera è meno di 100 Byte, verrà visualizzato un messaggio d'errore e il primo passo terminerà.

Se si verifica un errore di I/0 nella lettura del file di input, verrà visualizzato un messaggio di errore. L'assemblaggio terminerà dopo ogni errore di questo tipo.

Durante il secondo passo, il file sorgente è nuovamente letto, mentre il file programma viene scritto su disco. È possibile produrre su disco o stampare un listato dell'assemblaggio. Il file programma prodotto può poi essere caricato nel PET/CBM come ogni altro programma, perché usa il formato standard dei file PET/CBM. Tuttavia, proprio per questo, il file deve

essere in un unico blocco contiguo, senza "buchi".

La lista prodotta dall'assembler indicherà le locazioni di memoria e il codice oggetto in esadecimale, i numeri di linea del file sorgente e le linee originali del file sorgente (etichetta, comando, operando e commento). I numeri di linea corrispondono direttamente con i numeri di linea visualizzati dall'editor nell'editing del file sorgente. Questi numeri quindi possono essere usati per localizzare velocemente specifiche linee.

L'assembler si aspetta di trovare il formato di ogni linea di codice con quattro campi: una etichetta opzionale, un comando obbligatorio, un operando opzionale ed un commento. In alternativa, una linea può essere tutto commento se il primo carattere è un asterisco.

Le etichette possono avere da uno a sei caratteri. Il primo carattere deve essere una lettera, gli altri possono essere cifre o lettere. Il valore di una etichetta sarà l'indirizzo del prossimo byte di memoria definito dal comando o il valore assegnato all'etichetta mediante una direttiva. Tutti i valori delle etichette sono calcolati come valori a 16 bit senza segno e mantenuti come tali nella tavola dei simboli. Se una etichetta è definita due volte, verrà usata la prima definizione a tutte le definizioni doppie saranno indicate nella tavola dei simboli. Questi errori vengono conteggiati e, assieme agli altri, riportati alla fine dell'assemblaggio.

Il comando deve consistere di una istruzione mnemonica seguita da un indirizzo appropriato o da una direttiva. Le istruzioni mnemoniche sono quelle standard usate dalla MOS Technology. Per accelerare la velocità di assemblaggio, alle istruzioni mnemoniche è aggiunto un indirizzo invece di essere codificato nell'operando.

La lista seguente riporta i suffissi usati per ogni modalità di indirizzamento:

blank	assoluto, implicito o relativo
A	accumulatore
#	immediato
@	indiretto

```
940 IF S=0 THEN 1040
950 FOR X=1 TO 5: PRINT#204,LS$;"  ": NEXT
960 PRINT#204,"SIMBOLO    ESA-VALORE-DECIMALE": PRINT#204,LS$"  ";E$:
970 B$="":S$(0)="_": FOR E=1 TO S: P=0:FOR T=1 TO S:IF S$(T) [illegible] S$(P) THEN P=1
980 NEXT
990 IF B$< >S$(P) THEN 1010
1000 PRINT#204,"******  DEFINIZIONE MULTIPLA": EN=EN+1: GOTO 1030
1010 N=V(P): D$=STR$(N): GOSUB 1460: B$=S$(P)
1020 PRINT#204,B$;SPC(11-LEN(B$));H$;SPC(14-LEN(D$));D$
1030 S$(P)="_": NEXT
1040 PRINT#204,LS$"  ": PRINT#204,EN;" ERRORI"
1050 IF PR$="D" THEN PRINT#204,".END"
1060 PRINT"ASSEMBLAGGIO COMPLETO, ";EN;"ERRORI": GOTO 1550
1070 REM ==============================
1080 REM *** SUBROUTINE NIDIFICATE***
1090 REM ==============================
1100 M$=LEFT$(C$,3): A$=MID$(C$,4)
1110 RESTORE: FOR N=1 TO 56: READ I$: IF M$ >LEFT$(I$,3) THEN NEXT: RETURN
1120 PS=P: PB=VAL(MID$(I$,4,1)): IF A$="" THEN P=P+PB: RETURN
1130 P=P+2: IF A$="X" OR A$="Y" THEN P=P+1: RETURN
1140 IF A$="A" THEN P=P-1: PB=1
1150 RETURN
1160 M$=D$: FOR X=1 TO LEN(D$): I$=MID$(O$,X,1)
1170 IF I$="+" OR I$="-" OR I$="/" THEN 1190
1180 NEXT: GOSUB 1300: RETURN
1190 N=0: A$=MID$(M$,X+1)
1200 IF X>1 THEN O$=LEFT$(M$,X-1): GOSUB 1300: IF N [illegible] THEN 1290
1210 N1=N: O$=A$: GOSUB 1300: IF N<0 THEN 1290
1220 O$=M$: IF I$="+" THEN N2=N+N1
1230 IF I$= "-" THEN N2=N1-N
1240 IF I$= "/" THEN N2=INT(N1/N): IF LEFT$(M$,1) [illegible] THEN 1290
1250 N=N2
1260 IF N<0 THEN N=N+65536: GOTO 1260
1270 IF N>65535 THEN N=N-65536: GOTO 1270
1280 RETURN
1290 O$=M$: GOTO 1370
1300 IF O$<"A" THEN 1330
1310 FOR X=0 TO S: IF S$(X)=O$ THEN N=V(X): RETURN
1320 NEXT: GOTO 1370
1330 IF LEFT$(O$,1)< >"$" THEN 1380
1340 N=0: FOR X=2 TO 6: H$=MID$(O$,X,1): IF H$="" THEN RETURN
1350 X1=VAL(H$): IF X1 OR H$="0" THEN N=N*16+X1: NEXT
1360 X1=ASC(H$): IF X1>64 AND X1<71 THEN N=N*16+(X1-55): NEXT
1370 N=-1: RETURN
1380 N=INT(VAL(O$)): IF N OR O$="0" THEN RETURN
1390 IF O$="*" THEN N=P: RETURN
1400 IF LEFT$(O$,1)="'" AND LEN(O$) 1 THEN N=ASC(MID$(O$,2)): RETURN
1410 GOTO 1370
1420 IF PF=0 THEN GOSUB 620
1430 PRINT#2,CHR$(N);: RETURN
1440 IF C$>"@" THEN X1=PS+1
1450 N=0: FOR X=1 TO P-X1: B$=B$+"00 ": GOSUB 1420: NEXT: GOTO [illegible]
1460 H$="": FOR X=3 TO 0 STEP-1: X1=INT(N/(16 [illegible])): N=N-(X1*(16 [illegible]
1470 IF X1>9 THEN X1=X1+7
1480 H$=H$+CHR$(X1+48): NEXT: H1$=LEFT$(H$,2): H2$=RIGHT$(H$,2): RETURN
1490 OPEN 1,8,2,DR$+":"+FL$+".SRC,S,R": GOTO 1510
1500 INPUT#1,L$,C$,O$,T$
1510 INPUT#15,EN$,EM$,ET$,ES$: IF EN$="00"THEN RETURN
1520 PRINT CHR$(147)CHR$(18)"ERRORE DI DISCO": PRINT
1530 PRINT EN$;" ";EM$,ET$;" ";ES$
1540 PRINT: PRINT: PRINTCHR$(18)"ASSEMBLAGGIO INTERROTTO": PRINT: PRINT
1550 CLOSE1: CLOSE 2: CLOSE 204: CLOSE 15: END
1560 REM ==============================
1570 REM *** TAVOLA DEI DATI DELLE ISTRUZIONI ***
1580 REM ==============================
1590 REM CONTIENE 56 DATI, 1 PER ISTRUZIONE
1600 REM OGNI DATO CONSISTE DI:
1610 REM      3 CARATTERI MNEMONICI
1620 REM      1 CIFRA LUNGHEZZA ISTRUZIONE
1630 REM      ESTENSIONI MULTIPLE DI 5 CARATTERI
1640 REM      PER OGNI MODO DI INDIRIZZAMENTO:
1650 REM          2 CARATTERI SUFFISSO MNEMONICO
1660 REM          3 CIFRE VALORE CODICE OPERATIVO
1670 REM LA TAVOLA TERMINA CON UN DATO NULLO
1700 DATA "ADC3# 105  109Z 101@X097@Y113ZX117X 125Y [illegible]
1710 DATA "AND3# 041  045Z 037@X033@Y049ZX053X [illegible]
1720 DATA "ASL3  014Z 006A 010ZX022X 030","BCC2 [illegible]","BCS [illegible]
1730 DATA "BEQ2  240","BIT3  044Z 036","BMI2  048","BNE [illegible]","BPL [illegible]
1740 DATA "BRK1  000","BVC2  080","BVS2  112","CLC [illegible]","CLD [illegible]
1750 DATA "CLI1  088","CLV1  184"
1760 DATA "CMP3# 201  205Z 197@X193@Y209ZX[illegible]
1770 DATA "CPX3# 224  236Z 228","CPY3# [illegible]
1780 DATA "DEC3  206Z 198ZX214X [illegible]
1790 DATA "EOR3#073 [illegible]
1800 DATA "INC3 [illegible]
1810 DATA "JMP3  076@ 108","JSR3 [illegible]
1820 DATA "LDA3# 169  173Z [illegible]
1830 DATA "LDX3# 162  174Z 166Y 190ZY[illegible]
1840 DATA "LSR3  078Z 070A 074ZX086X [illegible]","NOP1 [illegible]
1850 DATA "ORA3# 009  013Z 005@X001@Y017ZX[illegible]
1860 DATA "PHP1  008","PLA1  104","PLP1  040", [illegible]
1870 DATA "ROR3  110Z 102A 106ZX118X 126","RTI1 [illegible]
1880 DATA "SBC3# 233  237A 229@X225@Y241ZX24[illegible]
1890 DATA "SED1  248","SEI1  120","STA3  141Z [illegible]
1900 DATA "STX3  142Z 134ZY150","STY3 [illegible]
1910 DATA "TAY1  168","TAX1 [illegible]
```

Z pagina zero
X assoluto con registro X
Y assoluto con registro Y
@X indiretto con registro X
@Y indiretto con registro Y
ZX pagina zero con registro X
ZY pagina zero con registro Y

Solo quattro le direttive sono disponibili, principalmente per abbreviare i tempi di assemblaggio. Esse sono:

- .ADR *n*, che definisce un indirizzo costante da due byte nel formato standard; otto bit meno significativi, otto bit più significativi.
- .BY *n*, che definisce una costante da un byte di valore *n*; se *n* è maggiore di 255 il byte conterrà gli otto bit meno significativi (un byte) del valore binario specificato.
- LABEL=*n*, definisce una etichetta simbolica con valore *n*. Il valore *n* deve essere stato definito in precedenza nel primo passo dell'assemblaggio.
- *=*n*, posiziona il contatore di programma (*program counter*) al valore indicato, come numero a 16 bit; se è specificata una etichetta, le viene assegnato lo stesso valore di *n*, il primo byte della nuova area di memoria. Il valore di *n* deve essere stato precedentemente definito nel primo passo. Questo cambio deve essere usato prima di assemblare ogni istruzione, se viene usato. Questo perché il file programma prodotto deve essere in un blocco contiguo con un indirizzo di partenza ben noto. Se non viene specificato è zero.

Gli operandi con istruzioni e con direttive di assembler possono essere specificati in un certo numero di formati, indicati nella tavola 1. Inoltre, per ogni valore delle espressioni di tavola 2 sono ammessi tre operatori che utilizzano uno dei formati detti (decimale, esadecimale, ecc).

Tutti gli operandi sono calcolati come quantità a 16 bit. Se un comando richiede solo otto bit (come una istruzione di caricamento immediato dell'accumulatore) verranno usati solo gli otto bit meno significativi.

Il commento in una linea di istruzione può essere lungo fino a 25 caratteri, e si possono usare tutti i caratteri (ecetto quelli già menzionati. Una linea di solo commento può essere lunga fino a 50 caratteri.

Ogni errore di assemblaggio viene indicato nella lista stampata mediante una linea che indica il campo errato. Alla fine dell'assemblaggio, viene indicato il numero totale di errori. Inoltre, viene stampata una tavola dei simboli in ordine alfabetico, con il valore di ogni simbolo in esadecimale e in decimale.

Consigli

Questi sono alcuni consigli per un uso efficiente dell'assemblatore.

- Separate i programmi più grandi in sezioni logiche più piccole, che possano essere testate e assemblate direttamente. Ciò permetterà un lavoro più veloce.
- Definite i simboli più comuni, le costanti e le memorie di lavoro all'inizio di ogni programma. I simboli e le etichette sono introdotte nella tavola dei simboli nell'ordine definito e ricercate sequenzialmente nello stesso ordine.
- Usate etichette corte, se volete che ce ne stiano molte nella tavola.
- Controllate l'uso e la lunghezza dei commenti, perché possono aumentare di molto il tempo di lettura del file sorgente durante l'editing e l'assemblaggio.
- Ricordate che i numeri di linea nel listato dell'assemblatore e nel listato del file sorgente corrispondono direttamente con i numeri di linea visualizzati dall'editor nell'editare un file sorgente. Possono quindi essere usati per localizzare velocemente linee specifiche.
- I nomi dei file possono essere lunghi al max 12 caratteri. Viene aggiunto un suffisso di quattro caratteri per identificare il tipo di file:

.SRC file sorgente
.BAK file copia creato automaticamente
.LST file listato salvato su disco

Il nome del file finale generato dall'assemblatore non ha suffissi.

Risparmiate tempo!

Per evitare i fastidi e gli errori di una lunga battitura è disponibile un dischetto (per drive 2040, 3040 e 4040) contenente i due programmi descritti in questo articolo già registrati e verificati.
Lo potete ricevere contrassegno compilando questa cedola e pagando al postino 40.000 lire, iva, imballo e spese di spedizione comprese.

È un servizio di Personal Software

Indirizzare in busta chiusa a
PERSONAL SOFTWARE, Via Rosellini 12
20124 MILANO

Cognome e nome

Indirizzo

Cap, località

Inviatemi il disco con l'assemblatore per il PET/CBM pubblicato nel numero 2 di Personal Software
Pagherò al postino 40.000 lire.

Come farsi umiliare da un computer

Carlo Sintini

Il gioco del NIM consiste nel disporre un numero arbitrario di stecchini o di fiammiferi su tre righe (si può giocare anche con un numero maggiore di righe), e poi due giocatori a turno tolgono quanti fiammiferi vogliono, purché tutti appartenenti ad una stessa riga. Vince il giocatore che riesce a togliere l'ultimo fiammifero.

Il gioco fu lanciato da un vecchio film di successo (*L'anno scorso a Marienbad* di Alain Resnais) e qualche anno fa fu rispolverato da Martin Gardner (si veda il primo volume di *Enigmi e giochi matematici* Sansoni). Anche io me ne sono occupato in *Quiz e giochi matematici* (Longanesi) per ideare un computer realizzato con scatole di cerini che "impara" a giocare al NIM.

Il prof. Bouton dell'Università di Harvard scoprì una straordinaria strategia vincente per questo gioco, che rende imbattibile il giocatore che effettua la prima mossa, e quasi imbattibile (se il suo avversario non conosce la strategia vincente) il giocatore che muove per secondo.

Il numero di fiammiferi contenuto in ciascuna riga deve essere trasformato in forma binaria, i tre numeri binari devono essere incolonnati, e se la somma delle cifre di *ogni* colonna è zero o pari, la posizione è sicura (cioè garantisce la vincita a chi l'ha giocata) altrimenti è insicura.

Per esempio la posizione con 5, 6 e 7 fiammiferi è insicura perché

$$5 \equiv 101_2$$
$$6 \equiv 110_2$$
$$7 \equiv 111_2$$

e la somma della prima colonna è 3, che è dispari.

Invece la posizione con 5, 2 e 7 fiammiferi è sicura perché

$$5 \equiv 101_2$$
$$2 \equiv 10_2$$
$$7 \equiv 111_2$$

e sommando le colonne non si ottiene mai un risultato dispari.

Quindi per vincere al NIM è sufficiente fare in modo che dopo ogni propria mossa, la posizione sia sicura. Ne deriva che il primo a muovere, se gioca correttamente, ha la certezza di vincere.

Nel programma che vi propongo (realizzato per un PET ma facilmente adattabile a qualunque personal) potrete giocare al NIM contro il calcolatore: esso sarà imbattibile se muoverà per primo, e duro da battere se giocherà per

Spiegazione delle principali routine

1000-1050	Subroutine per stampare il tabellone di gioco
2000-2090	Subroutine per stabilire se la mossa del computer è sicura. In 2010 il numero di fiammiferi di ciascuna riga viene trasformato in un numero binario (sfruttando le DATA finali). Successivamente vengono effettuate le somme per colonne (aggiungendo uno zero iniziale per portare a 4 cifre il numero binario e renderlo compatibile con le DATA) e se i risultati sono tutti pari, la variabile F viene posta uguale a zero, mentre in caso contrario viene posta uguale ad uno.
3000-3130	Subroutine per l'esecuzione della mossa del computer. Le variabili A(K) rappresentano il numero di fiammiferi per ciascuna riga. In 3010 esse vengono trasformate in B(K) allo scopo di poter variare il loro valore nell'analisi della mossa, senza che le A(K) perdano il loro valore originario. Nelle linee 3020-3060 vengono tolti uno alla volta i fiammiferi dalle righe (operando sulle B(K)), ed ogni volta con la subroutine 2000 si controlla se la mossa è sicura. In caso affermativo (F=0) la mossa viene eseguita alla linea 3045, altrimenti alle righe 3090-3100 viene eseguita una mossa a caso tra quelle possibili.
4000-4050	Subroutine contenente il test per stabilire dopo ogni mossa (del giocatore o del calcolatore) qual'è il numero massimo di fiammiferi contenuti nel tabellone (M) e su quale riga (W) si trovano. Il test serve anche per stabilire quando il gioco è finito (M=0).

AVVERTENZA IMPORTANTE: per un errore in fase di impaginazione la successione dell'articolo « Come [illegible] il computer » è errata. La corretta segue [illegible]

secondo contro un avversario che non conosca la strategia vincente.

Nel riquadro sono spiegate le routine principali. Il resto del programma è abbastanza chiaro e non ha bisogno di particolari spiegazioni. Le linee 220-230 servono per fermare il programma dopo che il giocatore ha eseguito la propria mossa, per evitare che il computer esegua immediatamente la sua.

Tutto il programma sfrutta l'istruzione GET invece della INPUT, per evitare di premere il tasto RETURN dopo ogni battuta. Ciò però comporta la limitazione di 9 fiammiferi al massimo per ciascuna riga. Se volete aumentare il numero, basta sostituire la 120 con

120 INPUT A$(K)

ed eliminare lo spazio vuoto dopo la I nella stringa alla linea 1030, per infittire i fiammiferi e permettere loro di essere tutti contenuti in una sola linea.

Se volete usare più di 15 fiammiferi per riga occorre però aggiungere altre DATA e ribattere quelle esistenti con cinque cifre invece di quattro.

Infine, se volete portare a quattro le linee del tabellone, ricordatevi di modificare la linea 2030 con

2030 FOR T=1 TO 4: BB$(K) =
= MID$(B$(T), K, 1) : NEXT

e aggiungere alla linea 2080 AND S <> 14. Oltre, ovviamente a portare tutti i cicli FOR...NEXT da 3 a 4. Buon divertimento.

```
10 REM: CARLO SINTINI
20 CLR: PRINT CHR$(147)
30 PRINT TAB(14)"GIOCO DEL NIM"
40 FOR K=1 TO 40: PRINT"-";: NEXT K
50 PRINT"ORA DISPORREMO DEI FIAMMIFERI SU"
60 PRINT"TRE RIGHE."
70 PRINT"CIASCUNO DI NOI POTRA' TOGLIERE"
80 PRINT"QUANTI FIAMMIFERI VUOLE, MA DA UNA"
90 PRINT"SOLA RIGA. VINCERA' CHI TOGLIE"
95 PRINT"L'ULTIMO FIAMMIFERO."
100 FOR K=1 TO 40: PRINT"-";: NEXT K
105 FOR K=1 TO 3
110 PRINT"QUANTI FIAMMIFERI SULLA"K"' RIGA? ";
120 GET A$(K): IF A$(K)="" THEN 120
130 A(K)=VAL(A$(K))
140 IF A(K)<=0 THEN 120
145 PRINT A$(K)
150 NEXT K
155 GOSUB 1000
160 PRINT"OK - VUOI COMINCIARE PER PRIMO? ";
170 GET R$:IF R$="" THEN 170
175 PRINT R$
180 IF R$="S" THEN 250
190 IF R$="N" THEN 210
200 GOTO 170
210 REM: MUOVE IL PET
220 PRINT"POSSO MUOVERE? ";
230 GET R$: IF R$<>"S" THEN 230
235 PRINT R$
240 GOSUB 3000
250 REM: MUOVE IL GIOCATORE
260 PRINT"VUOI TOGLIERE DALLA RIGA 1, 2 O 3? ";
270 GET R$: IF R$="" THEN 270
275 PRINT R$
280 R=VAL(R$)
290 IF R<1 OR R>3 THEN 270
310 PRINT"QUANTI FIAMMIFERI? ";
320 GET R$: IF R$="" THEN 320
325 PRINT R$
330 IF A(R)-VAL(R$)<0 THEN GOSUB 1000: GOTO 260
340 A(R)=A(R)-VAL(R$)
350 GOSUB 4000
360 IF M=0 THEN PRINT"HAI VINTO TU!": GOTO 500
370 GOSUB 1000
380 GOTO 210
500 REM: RICHIESTA DI PROSEGUIMENTO
510 PRINT"VUOI GIOCARE ANCORA?"
520 GET R$: IF R$="" THEN 520
530 IF R$="S" THEN 10
540 PRINT"CIAO!": END
1000 REM: STAMPA TABELLONE
1010 PRINT CHR$(147)
1015 FOR K=1 TO 3
1017 PRINT K"' RIGA";
1020 IF A(K)=0 GOTO 1040
1030 FOR J=1 TO A(K)
1035 PRINT TAB(15)"I ";
1038 NEXT J
1040 PRINT: PRINT: NEXT K
1050 FOR K=1 TO 40
1055 PRINT"-";
1058 NEXT K: RETURN
2000 REM: ANALISI SICUREZZA DELLA MOSSA
2010 F=0: FOR K=1 TO 3: FOR J=0 TO B(K)
2015 READ B$(K): NEXT J: RESTORE: NEXT K
2020 FOR K=1 TO 4: S=0
2030 BB$(K)="0"+MID$(B$(1),K,1)+MID$(B$(2),K,1)
2035 BB$(K)=BB$(K)+MID$(B$(3),K,1)
2040 READ C$
2050 IF BB$(K)=C$ THEN RESTORE: GOTO 2070
2060 GOTO 2040
2070 FOR J=1 TO 4: S(J)=VAL(MID$(C$,J,1))
2075 S=S+S(J): NEXT J
2080 IF S=0 OR S=2 OR S=4 OR S=6 GOTO 2090
2085 IF S=8 OR S=10 OR S=12 GOTO 2090
2088 F=1: RETURN
2090 NEXT K: F=0: RETURN
3000 REM: DECISIONE E MOSSA
3010 FOR K=1 TO 3: B(K)=A(K): NEXT K
3020 Y=1
3030 IF B(Y)=0 THEN B(Y)=A(Y): GOTO 3050
3040 B(Y)=B(Y)-1: GOSUB 2000
3042 IF F<>0 GOTO 3030
3045 FOR K=1 TO 3: A(K)=B(K): NEXT K
3048 GOTO 3110
3050 IF Y=3 GOTO 3070
3060 Y=Y+1: GOTO 3030
3070 GOSUB 4000
3090 X=INT(M*RND(1)+1)
3100 A(W)=M-X
3110 GOSUB 1000: GOSUB 4000
3120 IF M<>0 THEN RETURN
3130 PRINT"HO VINTO IO'": GOTO 500
4000 REM: RICERCA NUM. MAX PER RIGA
4005 M=0: W=0
4010 FOR H=1 TO 3
4015 IF A(H)>M THEN M=A(H): W=W+1
4020 NEXT H
4030 RETURN
4050 REM: NUM.MAX=M, SULLA RIGA W
5000 DATA 0000,0001,0010,0011,0100,0101,0110,0111
5010 DATA 1000,1001,1010,1011,1100,1101,1110,1111
```

SOFT REBIT COMPUTER BANK

A DIVISION OF G.B.C.

PROGRAMMI PER IL SINCLAIR ZX81

Tutti i programmi sottoelencati sono registrati su cassetta
Se non è specificata la dicitura "1K", necessitano dell'espansione di memoria.
Sono marcate con asterisco le cassette che possono essere usate anche sullo ZX80 con ROM 8K

CODICE	NOME E DESCRIZIONE	PREZZO VENDITA
TF/0100-01	*** SEI GIOCHI IN INGLESE (1K)** ORBIT - SNIPER - METEORS - LIFE WOLF PACK - GOLF	13.000
TF/0100-02	**GIOCHI EDUCATIVI IN INGLESE** MATEMATICA - OPERAZIONI FRAZIONI DIVERSI GRADI DI DIFFICOLTA'	13.000
TF/0100-03	*** PROGRAMMI GESTIONALI IN INGLESE** AGENDA TELEFONICA - FINANZA PERSONALE - BLOCK NOTES	13.000
TF/0100-04	*** SEI GIOCHI IN INGLESE** LUNAR LANDING - TWENTY ONE - COMBAT SUB STRIKE - COBE BREAKER - MAYDAY	13.000
TF/0100-05	*** GIOCHI EDUCATIVI IN INGLESE (1K)** OPERAZIONI ELEMENTARI PER BAMBINI CON QUATTRO GRADI DI DIFFICOLTA'	13.000
TF/0100-10	**SCACCHI IN INGLESE** SI GIOCA CONTRO IL CALCOLATORE CON DIVERSI GRADI DI DIFFICOLTA	26.000
TF/0100-11	*** VU-CALC IN INGLESE** POTENTE STRUMENTO DI CALCOLO ADATTO A RISOLVERE DIVERSI PROBLEMI	26.000
TF/0100-12	**FANTASY GAMES IN INGLESE** GIOCHI DI FANTASIA PER TUTTI I GUSTI	26.000
TF/0101-02	*** GIOCO SCACCHI** QUATTRO LIVELLI DIFFICOLTA' - LIBERTA' DI DISPOSIZIONE PEZZI - SOLUZIONE PROBLEMI	26.000
TF/0101-04	**VISIZXCALC** POTENTE STRUMENTO DI CALCOLO ADATTO A RISOLVERE DIVERSI PROBLEMI	26.000
TF/0101-06	**UNDICI GIOCHI (1K)** DIVERTIMENTO E BUONI ESEMPI DI PROGRAMMAZIONE IN BASIC E LINGUAGGIO MACCHINA	17.000
TF/0101-08	**LABIRINTO TRIDIMENSIONALE** DIVERSI LIVELLI DI DIFFICOLTA' PROGRAMMAZIONE DI ALTO LIVELLO CON GRAFICA OTTIMA	17.000
TF/0101-10	**TRE GIOCHI SPECIAL (1K)** USATE IL SINCLAIR COME UN ORGANO VEDETE I BATTERI CHE SI RIPRODUCONO	17.000
TF/0101-12	*** GESTIONE PICCOLI ARCHIVI** GESTIONE COMPLETA DI PICCOLI ARCHIVI	17.000
TF/0101-14	*** SIMULATORE CUBO MAGICO TRIDIMENSIONALE** PER GLI APPASSIONATI DEL CUBO MENO FATICOSO DEL CUBO REALE	17.000
TF/0101-16	*** RISOLUTORE CUBO MAGICO** PER RISOLVERE IL CUBO IN POCO PIU' DI UN MINUTO	17.000
TF/0101-18	**'DEFENDER** UN PO' DI BRIVIDO CON IL SINCLAIR VELOCITA' ECCEZIONALE	17.000
TF/0101-20	**STAR-TREK** MISSIONE GALATTICA CON IMPREVISTI ED EMOZIONI QUATTRO LIVELLI DIFFICOLTA	17.000
TF/0101-22	**CENTIPEDE** PROVATE A DISTRUGGERE IL BRUCO CHE SI DIVIDE SE LO COLPITE - BRAVO CHI CI RIESCE	17 000
TF/0101-24	**ASTEROIDI** UN BUON PASSATEMPO PER VOI E PER I VOSTRI AMICI	17 000
TF/0101-26	**TIRANNOSAURO** PER CHI SI ANNOIA COL LABIRINTO - GRAFICA DINAMICA E TERRORE	17.000
TF/0101-28	**ZUC** GIOCO AFFASCINANTE PER UNO O DUE GIOCATORI NON USATELO TROPPO'	17.000
TF/0102-02	*** SETTE GIOCHI** BIORITMO - 21 - CONTO ALLA ROVESCIA - HAMMURABI ROULETE RUSSA - FUGA DAL CASTELLI - METEORITI	22.000
TF/0102-04	*** SETTE GIOCHI** MASTER-MIND - ORBITA - GOLF - BOMBARDAMENTO LANCIA MINE - SOS SOS - CAMMELLO	22.000
TF/0102-06	*** SETTE GIOCHI** ALLUNAGGIO - SLALOM - CACCIA SOTTOMARINA - ALIENI TIRO RAPIDO - ATTACCO MARZIANO - LA GRANDE RAPINA	22.000
TF/0102-08	*** SETTE GIOCHI** SUPER AVVENTURA - SOLITARIO - REVERSE - LABIRINTO ABBATTI IL MURO - GOLF - GIU' DENTRO	22.000
TF/0102-10	*** SETTE GIOCHI** BATTAGLIA NAVALE BUCHI NERI - ODISSEA - MEMORY ANAGRAMMI - ARMA GIOVIANA - TRENI IN CORSA	22.000
TF/0102-12	*** GESTIONE FINANZIARIA PERSONALE** POSSIBILITA DI MEMORIZZARE I CONTI SU NASTRO	22.000
TF/0102-14	*** AGENDE:** RUBRICA INDIRIZZI ARRICCHITA - ARCHIVIAZIONE NOTIZIE CON POSSIBILITA RICERCA	22.000
TF/0102-16	*** MATEMATICA E FISICA** FRAZIONI - STATISTICA - TEMPERATURE PROBLEMI - CONVERSIONI DI BASE	22.000
TF/0102-18	*** MATEMATICA, FISICA E VOCABOLARIO** SOMMARE DIVERTENDOSI - LA BILANCIA - CALCOLO DEI VOLUMI MOLTIPLICAZIONI - VOCABOLI	22.000
TF/0102-20	*** TOOL-KIT** STRUMENTO INDISPENSABILE AD OGNI PROGRAMMATORE CHE VOGLIA AFFINARE LE SUE ABILITA	22.000
TF/0103-00	**2 GIOCHI IN ITALIANO (2K)** MESSAGGI IN CORSA E BISCIA GIOCO DI ABILITA	17 000
TF/0103-02	**ISTO-CARATTERI (2K)** ISTOGRAMMI - INGRANDIMENTO DI CARATTERI TROVA MOLTE APPLICAZIONI	17.000
TF/0103-04	**DAMA e TOTOCALCIO** DIVERTIMENTO E INVITO ALLA FORTUNA	17.000
TF/0103-06	**RUBRICA** AGENDA TELEFONICA CONTIENE FINO A 200 INDIRIZZI	22.000
TF/0103-08	**3 GIOCHI IN ITALIANO** CODICE SEGRETO - BASE ALIENA - UFO	17 000
TF/0103-10	**TRATTAMENTO (16K) W.P.** ELABORAZIONE TESTI PER CONSERVARE E STAMPARE NOTIZIE E SCRITTI DI OGNI GENERE	[illegible]
TF/0103-12	**2 GIOCHI IN ITALIANO (16K)** AIUTO - BERSAGLIO	17.000
TF/0103-14	**3 GIOCHI IN ITALIANO (16K)** CANNONATE - TIRO A VOLO - SLALOM	17.000
TF/0103-16	**2 GIOCHI IN ITALIANO (16K)** GALASSIA - LABIRINTO	17.000
TF/0103-18	**2 GIOCHI (16K)** SCONTRO - FAR WEST	17.000
TF/0103-20	**2 GIOCHI (16K)** ROULETTE - PENSACI	17.000
TF/0103-22	**MISSILI OSTACOLI (16K)** ANCORA DUE DIVERTENTI GIOCHI PER ZX81	17 000

PROGRAMMI PER IL TRS-80 MOD. II

I programmi sottoelencati sono forniti su disco 8'

CODICE	NOME E DESCRIZIONE	PREZZO VENDITA
TF 4502-00	**INVENTORY CONTROL** 3000 ARTICOLI DI MAGAZZINO - 200 FORNITORI S SCORTA - DIVISIONE IN CLASSI - STATISTICHE	345 000
TF/4506-00	**MAILING LIST** 3000 NOMI E INDIRIZZI IN FORMATO COMPATTO 2000 IN FORMATO ESPANSO - SELEZIONI E STAMPE	140 000
TF/4507 00	**MAILING LIST II (RICHIEDE 2 DISK)** COME IL MAILING LIST MA SE USATO CON LO SCRIPSIT PERMETTE LA STAMPA DI CIRCOLARI SELEZIONATE	210.000
TF/4512-00	**VERSA FILE II** CREATEVI IL VOSTRO SISTEMA DI CLASSIFICAZIONE AUTOMATICA DELLE INFORMAZIONI - FACILE DA USARE	125.000
TF 4511 00	**VISICALC II** SUPERPROGRAMMA CHE GESTISCE COMPLESSE PROIEZIONI E GRANDI QUANTITA DI DATI PER SIMULAZIONI	420.000
TF 4510-00	**PROFILE II** GESTIONE DI MOLTI DATI CON MOLTI CRITERI DI SELEZIONE COLLEGAMENTO ALLO SCRIPSIT - STAMPE	340.000
TF 4530-00	**SCRIPSIT II** UNO DEI SISTEMI DI GESTIONE DEI DATI FRA I PIU POTENTI SUL MERCATO	620.000
TF 4540-00	**STATISTICAL ANALISYS** STATISTICHE - VARIANZE - COVARIANZE - ISTOGRAMMI CORRELAZIONI FREQUENZE - ECC	180 000
TF 4701-00	**FORTRAN** STANDARD ANSI-66 - EDITORE - COMPILATORE EDITORE DI LINEA - BIBLIOTECA SOTTOPROGRAMMI	520.000
TF 4702-00	**EDITOR/ASSEMBLER** EDITORE - MACROASSEMBLER - EDITORE DI LINEA BIBLIOTECA FORTRAN - TABELLA CORRISPONDENZE	350 000
TF 4703-00	**COBOL** VERSIONE ESPANSA ANSI-74 - ISAM MULTICHIAVE ACCEPT DISPLAY - DEBUG MODULO RUN-TIME	520 000

Prezzi netti IVA esclusa

TF/4704-00	**COBOL RUN-TIME** PER L'ESECUZIONE DI PROGRAMMI SCRITTI E COMPILATI COL COBOL COMPILER	600.000
TF/4705-00	**BASIC COMPILER** ISAM MONOCHIAVE - 14 CIFRE DI CALCOLO MODULO RUN-TIME - NON COMPATIBILE COL BASIC INTERPRETE	430.000
TF/4706-00	**BASIC RUN-TIME** PER L'ESECUZIONE DI PROGRAMMI SCRITTI E COMPILATI COL BASIC COMPILER	[illegible]
TF/4710-00	**TEXT EDITOR** SI PUO' INTEGRARE IN OGNI LINGUAGGIO DEL MOD 2 RICERCHE E SOSTITUZIONI GLOBALI PIU' ALTRO	150.000
TF/4714-00	**REFORMATTER (RICHIEDE 2 DISCHI)** SCRITTURA - LETTURA - TRASFERIMENTO DI ARCHIVI TRA DISCHI TRSDOS E DISCHI IMB 3741/3742	450.000

PROGRAMMI PER IL TRS-80 MOD. III VERSIONE DISCO

La minima configurazione per l'uso dei programmi presentati è indicata a fianco del nome
Tutti i programmi sono in inglese

CODICE	NOME E DESCRIZIONE	PREZZO VENDITA
TF/1508-00	**IN-MEMORY INFORMATION (16K)** CLASSIFICAZIONE DELLE INFORMAZIONI SALVATAGGIO E RICERCA SU DISCO	36.000
TF/1551-00	**DISK MAILING LIST** PIU' POTENTE DELLA VERSIONE SU CASSETTA	70.000
TF/1553-00	**INVENTORY CONTROL (32K 2 DISCHI)** FINO A 1000 ARTICOLI CON RAPPORTI SULLE VENDITE E LE ROTAZIONI DEL MAGAZZINO	170.000
TF/1558-00	**BUSINESS MAILING LIST (32K 2 DISCHI)** FINO A 990 NOMI - CON 48K E 4 DISCHI 2970 NOMI)	170.000
TF/1559-00	**MANUFACTURING INVENTORY CONTROL (32K 2 DISCHI)** GESTIONE DELLE DISTINTE BASE - 20 PRODOTTI FINITI e 1900 MATERIE PRIME PER DISCO	320.000
TF/1562-00	**PROFILE (32K 1 DISCO)** GESTIONE DI ARCHIVI CON RICERCHE MULTIPLE ARCHIVI ACCESSIBILI DA PROGRAMMI UTENTE	135.000
TF/1563-00	SCRIPSIT DISK (32K 1 DISCO) PROCEDURA DI TRATTAMENTO DELLA PAROLA STAMPE MULTIPLE - FACILE EDITING	150.000
TF/1565-00	**MICROFILES (32K 1 DISCO)** VERSIONE SOFISTICATA DEL PROFILE VELOCISSIMO - COMANDI A SINGOLO TASTO	165.000
TF/1567-00	**VISICALC MOD. 3 (32K 1 DISCO)** SUPERPROGRAMMA CHE PERMETTE DI LAVORARE CON PROIEZIONI E MODELLI DI SIMULAZIONE	175.000
TF/1569-00	**VISICALC AVANZATO MOD 3 (32K 1 DISCO)** UNISCE ALLA POTENZIALITA DEL VISICALC L ENORME FLESSIBILITA DEL MOD 3	300.000
TF/1603-00	**PERSONAL FINANCE DISK (16K)** FORNITO IN VERSIONE CASSETTA PUO' ESSERE ADATTATO AL DISCO (FINO A 32K 2 DISCHI)	35.000
TF/2010-00	**DISK BASIC COURSE (16K 1 DISCO)** UN GRANDE CORSO SU 4 DISCHI CON TUTTE LE PIU POTENTI ISTRUZIONI DEL BASIC MOD 3	60.000
TF 1604-00	**VERSAFILE (32K 1 DISCO)** SCRIVETE CIO CHE VI VIENE IN MENTE E IL TRS 80 LO RICORDA - CHIEDETEGLIELO!	50.000
TF/2201-00	**FORTRAN (32K 2 DISCHI)** COMPILATORE - EDITORE DI TESTI - EDITORE DI LINEA - LIBRERIA	160.000
TF 2202-00	**EDITOR/ASSEMBLER DISK (32K 2 DISCHI)** ASSEMBLATORE - EDITORE DI TESTI EDITORE DI LINEA - TABELLA DELLE CORRISPONDENZE	160.000
TF/2204-00	**BASIC COMPILER (48K 2 DISCHI)** TUTTA LA POTENZA DEL LINGUAGGIO MACCHINA DAL BASIC - INCOMPATIBILE CON IL BASIC INTERPRETE	280.000

PROGRAMMI PER IL TRS-80 MOD. III VERSIONE CASSETTA

La minima configurazione per l'uso dei programmi presentati è indicata a fianco del nome
Tutti i programmi sono in inglese

CODICE	NOME E DESCRIZIONE	PREZZO VENDITA
TF/1502-00	**IN-MEMORY PROGRAM (16K)** CLASSIFICAZIONE DELLE INFORMAZIONI SALVATAGGIO E RICERCA	32.000
TF/1503-00	**MAILING LIST (16K)** GESTIONE INDIRIZZI CON STAMPA ETICHETTE - 80 NOMI PER VOLTA OGNI 16K	35.000
TF/1505-00	**SCRIPSIT (16K)** PROGRAMMA COMPLETO DI TRATTAMENTO DEI TESTI - MOLTO POTENTE	120.000
TF 1602-00	**PERSONAL FINANCE (4K)** GESTION ENTRATE E USCITE FAMILIARI GESTIONE BILANCIO MENSILE	30.000
TF/1603-01	**PERSONAL FINANCE DISK (16K)** FORNITO IN VERSIONE CASSETTA PUO ESSERE ADATTATO AL DISCO (FINO A 32K 2 DISCHI)	35.000
TF 1605-00	**ASTROLOGY (16K)** PRODUZIONE DI OROSCOPI PERSONALI SE COLLEGATO AD UNA STAMPANTE PRODUCE IL QUADRO ASTRALE	50.000
TF/1701-00	**MATHEMATIC COURSE (4K)** INSEGNA AI BAMBINI LE 4 OPERAZIONI	37.000
TF/1702-00	**ALGEBRA COURSE (4K)** IMPARARE L'ALGEBRA È FACILE! - E NON È NECESSARIO ASPETTARE DI FREQUENTARE LE MEDIE!	30.000
TF/1703-00	**STATISTIC COURSE (16K)** PER IMPARARE AGEVOLMENTE E FACILMENTE AD USARE LE TEORIE STATISTICHE ANCHE PER GRANDI	50.000
TF/1705-00	**ADVANCED STATISTICS (16K)** INTEGRA E COMPLETA IL CORSO DI STATISTICA CON QUALCOSA DI PIU COMPLESSO	80.000
TF/1706-00	**I.Q. BUILDING (16K)** CALCOLO E MIGLIORAMENTO DEL PROPRIO QUOZIENTE D'INTELLIGENZA TRAMITE SEMPLICI TEST	50.000
TF/1712-00	**SHOW & SPELL (16K)** FACILE CORSO DI GRAMMATICA INGLESE PER BAMBINI	60.000
TF/2000-00	**DEBUG (16K)** PROGRAMMA DI CONTROLLO E DI ESECUZIONE PER PROGRAMMI IN LINGUAGGIO MACCHINA IN MEMORIA	40.000
TF/2001-00	**T-BUG (16K)** CARICA UN PROGRAMMA IN LINGUAGGIO MACCHINA DA CASSETTA E NE PERMETTE IL DEBUG	35.000
TF/2002-00	**EDITOR-ASSEMBLER (16K)** PERMETTE D'INTRODURRE UN PROGRAMMA IN LINGUAGGIO SIMBOLICO ZILOG E DI ASSEMBLARLO	50.000
TF/2003-00	**LEVEL 1 COURSE (4K)** CORSO DI BASIC LIV 1	30.000
TF/2005-00	**BASIC COURSE LEVEL 2 PT.1 (16K)** CORSO DI BASIC ELEMENTARE - PRIMA PARTE	30.000
TF/2006-00	**BASIC COURSE LEVEL 2 PT. 2 (16K)** CORSO DI BASIC ELEMENTARE - SECONDA PARTE	35.000
TF/2009-00	**TINY PASCAL TAPE (16K)** COMPILATORE DI UN SUBSET DEL LINGUAGGIO PASCAL - POTENZIALITA' MAI VISTA!	38.000

PROGRAMMI PER IL TRS-80 POCKET COMPUTER

Tutti i programmi sono forniti su cassetta e sono in inglese

CODICE	NOME E DESCRIZIONE	PREZZO VENDITA
TF/3511-00	**CIVIL ENGINEERS** PROGRAMMI DI INGEGNERIA - CALCOLO TELAI - SFORZI AI BULLONI - TRAVI INCASTRATE - ECC	42.500
TF/3513-00	**AVIATION** CALCOLO DEL PIANO DI VOLO - ANGOLO DI DERIVA CONVERSIONI TRA UNITA' DI MISURA - ECC	42.500
TF/3514-00	**MATH DRILL** ESERCIZI PER GLI SCOLARI DELLE PRIME CLASSI POSSIBILITA' DI INTRODURRE NUOVI PROBLEMI	38.000
TF/3515-00	**GAMES ONE** CANNIBALI E MISSIONARI - NIM - ATTERRAGGIO NELLO SPAZIO - CACCIA AL TESORO - ECC	38.000
TF/3516-00	**BUSINESS MARKETING** METODO DELLA MEDIA MOBILE PER IL CALCOLO E LA CORREZIONE AUTOMATICA DELLE PREVISIONI - ECC	35.000
TF/3517-00	**BUSINESS FINANCE** SETTE PROGRAMMI DIFFERENTI PER AIUTARE L'UOMO D'AFFARI - CALCOLI INTERESSI - GIORNI - ECC	35.000
TF/3518-00	**PERSONAL FINANCE** GESTIONE DEL BILANCIO FAMILIARE - GESTIONE C/C BANCARIO - INTERESSI - CONVERSIONI - ECC	35.000

PROGRAMMI PER IL TRS-80 COLOR COMPUTER

Tutti i programmi sono distribuiti sotto forma di CARTRIDGE (memoria allo stato solido)
Tutti i programmi sono in inglese
Tutti i programmi contrassegnati da asterisco richiedono l'uso di joystick

CODICE	NOME E DESCRIZIONE	PREZZO VENDITA
TF/3019-00	**ROM DIAGNOSTICA** CONTROLLO DELLA PERFETTA EFFICIENZA DEL VOSTRO CACOLATORE	39.000
TF/3050-00	**SCACCHI** DA ALLENAMENTO, MA ANCHE DA COMBATTIMENTO!	90.000
TF/3051-00	*** QUASAR COMMANDER** RADAR - PILOTA AUTOMATICO - CAMPI DI FORZA DIVERSI LIVELLI DIFFICOLTA'	60.000
TF/3052-00	*** PINBALL** IL CLASSICO GIOCO DEL FLIPPER ORA ANCHE SUL TELEVISORE - DA 1 A 4 GIOCATORI	60.000
TF/3055-00	**CHECKERS** GIOCO DELLA DAMA A DUE LIVELLI DI DIFFICOLTA' PREVEDE LE 3 MOSSE SUCCESSIVE	60.000
TF/3056-00	*** SUPER BUSTOUT** GIOCO RAPIDO PER 1-4 GIOCATORI - SINGOLO O IN EQUIPE - SFONDATE LE LINEE COL PALLONE	60.000
TF/3057-00	*** DINO WARS (16K CONSIGLIATI)** DUE GIOCATORI ALLE PRESE CON I DINOSAURI GRAFICA E SONORO REALISTICI	70.000
TF/3058-00	*** SKIING (16K CONSIGLIATI)** DISCESA SCHISTICA CONTRO IL TEMPO VISTA CON GLI OCCHI DELLO SCIATORE	70.000
TF/3059-00	*** COLOR BACKGAMMON** CLASSICO GIOCO DI SOCIETA' - CONTRO IL CLACOLATORE O UN ALTRO AVVERSARIO	60.000
TF/3060-00	*** SPACE ASSAULT** GLI EXTRATERRESTRI VI INVADONO LO SCHERMO E VI ATTACCANO! - BUONA FORTUNA!	50.000
TF 3061-00	*** ART GALLERY (16K CONSIGLIATI)** CREATE LA VOSTRA GALLERIA DI QUADRI MODERNI - CONSIGLIATI I JOYSTICK	80.000
TF/3063-00	*** PROJECT NEBULA** RESPINGETE GLI INVASORI DELLA VOSTRA GALASSIA - 4 LIVELLI - APPASSIONANTE!	90.000
TF/3103-00	**COLOR FILE** PICCOLO SISTEMA DI GESTIONE PER TANTI ARCHIVI - SI USA COL REGISTRATORE A CASSETTE	60.000
TF/3101-00	**PERSONAL FINANCE** PIANIFICATE IL BUDGET FAMILIARE COMPARATE ENTRATE E USCITE - PREVEDETE IL BILANCIO	90.000
TF/3151-00	*** BINGO MATH** INSEGNA LE 4 OPERAZIONI E IL RICONOSCIMENTO DEI NUMERI - 1-2 GIOCATORI	60.000
TF/3152-00	**TYPING TUTOR** ESERCIZI BASATI SU LETTERE E PAROLE CONTROLLA VELOCITA - RIFLESSI - ERRORI	60.000

Prezzi netti IVA esclusa

TF/3153-00	**LEARNING LAB** COMBINAZIONE DI LOGICA E TESTI PER INSEGNARE IL COLOR BASIC - ORGANIZZAZIONE E STESURA	80.000
TF/3154-00	**HANDY MAN** CALCOLO DELLE ESATTE NECESSITA' DEL LAVORO DEL BRICOLAGE - MATERIALI - CONSIGLI	60.000

PROGRAMMI PER IL BMC IF 800 MOD. 20

Tutti i programmi sottoelencati sono forniti su disco 5"

CODICE	NOME E DESCRIZIONE	PREZZO VENDITA
TF/2502-00	**FORTRAN-80 (RICHIEDE IL CP/M)** EDITORE - COMPILATORE - EDITORE DI LINEA ANSI-66	800.000
TF/2504-00	**BASIC COMPILER (RICHIEDE IL CP/M)** RENDE PIU' VELOCI I PROGRAMMI IN BASIC INTERPRETE	650.000
TF/2506-00	**MBASIC (RICHIEDE IL CP/M)** BASIC INTERPRETE	300.000
TF/2508-00	**T-MAKER 2 (RICHIEDE IL CP/M)** GESTIONE DI TESTI E ARCHIVI IN COMBINAZIONE CON TUTTI I TIPI DI CALCOLO NUMERICO	700.000
TF/2510-00	**SUPERCALC (RICHIEDE IL CP/M)** IL VOSTRO FOGLIO ELETTRONICO A COLORI CALCOLI E PRVISIONI FINANZIARIE	500.000
TF/2512-00	**WORD STAR (RICHIEDE IL CP/M)** L'ULTIMO E IL PIU' PERFEZIONATO PROGRAMMA PER GESTIONE DI TESTI - PUO' TUTTO!	800.000
TF/2514-00	**WORD INDEX (RICHIEDE IL CP/M)** IN ABBINAMENTO AL WORD STAR PERMETTE LE STAMPE DI MANUALI - INDICE E RIASSUNTI AUTOMATICI	300.000
TF/2516-00	**COBOL-80 (RICHIEDE IL CP/M)** COMPILATORE ANSI-74 - ACCEPT/DISPLAY - EDITORE	1.300.000
TF/2518-00	**DBMS (RICHIEDE IL CP/M)** GESTIONE COMPLETA DI GRANDI ARCHIVI RICERCHE MULTICHIAVE - STAMPE DI TUTTI I TIPI	1.000.000
TF/2520-00	**ARCHIVI (IN OKI-BASIC)** IL DISCO CONTIENE DIVERSI PROGRAMMI DI ARCHIVIO PIU' UN DEMO E UN PROGRAMMA TYPEWRITER	400.000

PROGRAMMI PER IL COMMODORE (LINEA 3000 - 4000 - 8000)

Tutti i programmi sottoelencati sono forniti su disco 5"
Per ogni programma verrà specificato il modello

CODICE	NOME E DESCRIZIONE	PREZZO VENDITA
TF/1102-00	**FATTURAZIONE MANUALE (3000)** GESTIONE CLIENTI - EMISSIONE FATTURE E TRATTE - SENZA CODIFICA MAGAZZINO	700.000
TF/1104-00	**GESTIONE CONDOMINII (3000)** GESTIONE DI PIU' SCALE - EMISSIONE AUTOMATICA LETTERE - CIRCOLARI - SOLLECITI	800.000
TF/1106-00	**GESTIONE CONDOMINII (4000)** GESTIONE DI PIU' SCALE - EMISSIONE AUTOMATICA LETTERE - CIRCOLARI - SOLLECITI	800.000
TF/1108-00	**GESTIONE COMDOMINII (8000)** GESTIONE DI PIU' SCALE - EMISSIONE AUTOMATICA LETTERE - CIRCOLARI - SOLLECITI	800.000
TF/1110-00	**WORD PROCESSOR (8000)** PROCEDURA COMPLETA DI TRATTAMENTO DEI TESTI PERMETTE CIRCOLARI SELEZIONATE	630.000
TF/1112-00	**ASSEMBLER (3000)** EDITORE - ASSEMBLATORE SIMBOLICO 6502	115.000
TF/1114-00	**PASCAL (3000)** SUBSET UCSD PASCAL - COMPILATORE - EDITORE	115.000
TF/1116-00	**GESTIONE LABORATORI ANALISI MEDICHE (3000)** GESTIONE COMPLETA DI UN LABORATORIO - STAMPA I DOCUMENTI PER GLI ENTI - STATISTICHE	900.000
TF/1118-00	**GESTIONE LABORATORI ANALISI MEDICHE (4000)** GESTIONE COMPLETA DI UN LABORATORIO STAMPA I DOCUMENTI PER GLI ENTI - STATISTICHE	900.000
TF/1120-00	**GESTIONE LABORATORI ANALISI MEDICHE (8000)** GESTIONE COMPLETA DI UN LABORATORIO STAMPA I DOCUMENTI PER GLI ENTI - STATISTICHE	900.000
TF/1122-00	**VISICALC (4000 + ROM AGGIUNTIVA FORNITA)** SUPERPROGRAMMA PER GESTIONE DATI NUMERICI PROIEZIONI - SIMULAZIONI	310.500
TF/1124-00	**VISICALC (8000 + ROM AGGIUNTIVA FORNITA)** SUPERPROGRAMMA PER GESTIONE DATI NUMERICI PROIEZIONI - SIMULAZIONI	310.500
TF/1126-00	**COM-PLUS (8000)** UTILE ACCESSORIO PER SUPERARE LA BARRIERA DELL'INCOMPATIBILITA' TRA I DIVERSI SISTEMI	50.000
TF/1128-00	**WORD-CRAFT (8000 + CHIAVE D'ACCESSO)** ALTRA VERSIONE DI WORD PROCESSOR CON CARATTERISTICHE ADERENTI AD ESIGENZE DIVERSE	632.500
TF/1130-00	**VIGIL (3000)** LINGUAGGIO ORIENTATO ALLA PRODUZIONE DI GIOCHI SONORI E GRAFICI - 9 GIOCHI ESEMPIO FORNITI	120.000

PROGRAMMI PER IL VIC-20 CBM

Tutti i programmi sottoelencati sono registrati su cassetta
Se non specificato, si intende che i programmi funzionano con la memoria in configurazione base

CODICE	NOME E DESCRIZIONE	PREZZO VENDITA
TF/9350-00	**VICALCOLO (RAM-STANDARD)** CALCOLI MATEMATICI - INTERESSE COMPOSTO PIANI DI AMMORTAMENTO	40.000
TF/9350-02	**CREO-LISTA-STAMPA (RAM STANDARD)** PER REALIZZARE TESTI, LETTERE E CIRCOLARI PERSONALIZZATE	40.000
TF/9350-04	**DAMA** (RAM STANDARD) GIOCO DELLA DAMA CONTRO IL CALCOLATORE PER TUTTI, GRANDI E PICCOLI	17.500
TF/9402-00	**THE ALIEN WITH JOYSTICK (6K)** PROVATE A CALARVI NEI PANNI DELL'ALIENO!	60.000
TF/9404-00	**AMOK** UN GIOCO DI COMBATTIMENTO E DI VIOLENZA	60.000
TF/9406-00	**THE ALIEN** SIETE L'ALIENO E DOVETE SOPRAVVIVERE!	60.000
TF/9408-00	**3-D MAZE** TROVATE L'USCITA DAL LABIRINTO TRIDIMENSIONALE! DIVERSI LIVELLI DI DIFFICOLTA'	36.000
TF/9410-00	**ALIEN BLITZ (JOYSTICK OPZIONALE)** DISTRUGGETE GLI INVASORI DEL CIELO!	60.000
TF/9412-00	**VICAT** GESTIONE DI UN ARCHIVIO SEQUENZIALE SU CASSETTA	60.000
TF/9300-00	**CASSETTA PROGRAMMI DIMOSTRATIVI** DIMOSTRA LA POTENZA DEL VIC	15.700

PROGRAMMI PER IL VIC-20 CBM

Tutti i programmi sottoelencati sono registrati **su cartridge**
Se non specificato, si intende che i programmi funzionino con la memoria in configurazione base

CODICE	NOME E DESCRIZIONE	PREZZO VENDITA
TF/9300-04	**INVASORI SPAZIALI** GRANDE REALISMO - ALTA VELOCITA' NON VI FATE PRENDERE DAL PANICO!	37.000
TF/9300-06	**GARA AUTOMOBILISTICA** PROVATE L'EBREZZA DELLA VELOCITA' E DELLA COMPETIZIONE - 1 o PIU' GIOCATORI	37.000
TF/9300-08	**ATTERRAGGIO SU GIOVE** ESSERE AL COMANDO DI UNA ASTRONAVE NON È SEMPLICE MA QUESTO LO IMPARERETE A VOSTRE SPESE	37.000
TF/9300-10	**GIOCO DEL POKER** ATTENZIONE! - POTRESTE RESTARE POVERI! QUI NON SI TRATTA DI FORTUNA	37.000
TF/9300-12	**IL FANTASMA DI MEZZANOTTE** FUGGITE VIA DALLA CASA INFESTATA DAGLI SPIRITI SE VE NE RIMANE IL TEMPO	37.000
TF/9300-14	**BILANCIO FAMILIARE** PIANIFICATE LE VOSTRE SPESE IN FUNZIONE DELLE ENTRATE GESTITE IL VOSTRO C/C BANCARIO	37.000
TF/9300-16	**APPLICAZIONI MATEMATICHE** UN VALIDO AIUTO TESO AL MIGLIORAMENTO DELLE PROPRIE CAPACITA' DI CALCOLO	37.000
TF/9300-18	**SLOT MACHINE** IL CELEBRE GIOCO D'AZZARDO	37.000
TF/9300-20	**AVENGER** INTERESSANTE GIOCO DI SIMULAZIONE	37.000
TF/9300-22	**RAT RACE**	37.000
TF/9300-24	**MOLE ATTACH**	37.000
TF/9300-26	**SCACCHI**	37.000
TF/9300-28	**MATHEMATICAL ANALYSIS VIC-GRAF.** VALIDO AIUTO NELLO STUDIO DI COMPLICATE EQUAZIONI E FUNZIONI E DEI RELATIVI GRAFICI	95.000
TF/9300-30	**VIC FORTH LANGUAGE** CARTRIDGE PER PROGRAMMARE IL VIC CON IL NUOVO E POTENTE LINGUAGGIO FORTH INTERESSANTISSIMO A CORREDO, MANUALE PER IL RAPIDO APPRENDIMENTO	95.000
TF/9300-32	**MATHEMATICAL ANALYSIS VIC-STAT.** ROM IN LINGUAGGIO ASSEMBLER PER SEMPLIFICARE IL LAVORO CON STATISTICHE E GRAFICI SUL VIDEO CON UNA SOLA ISTRUZIONE SI OTTENGONO ISTOGRAMMI DEVIZIONI, STANDARD VARIANZE ECC	95.000

PROGRAMMI PER APPLE II

Tutti i programmi sono forniti su disco
Per ogni programma è indicata la lingua (italiano-inglese) in cui è stato scritto
Ove non indicato, si intende che i programmi girano sulla configurazione 16K 1 disco

CODICE	NOME E DESCRIZIONE	PREZZO VENDITA
TF/5502-00	**TOTOCALCIO SISTEMA A CORREZIONE D'ERRORI (It.)** ELABORAZIONE DI SISTEMI RIDOTTI	80.000
TF/5504-00	**TOTOCALCIO CHIAVE ALFA 6 SUPER (It.)** SISTEMA RIDOTTO 10 TRIPLE CON FATTORE DI RIDUZIONE - INDICATO AL SISTEMISTA SERIO	70.000
TF/5506-00	**TOTOCALCIO SISTEMA DERIVATO A ROTAZIONE (It.)** ELABORAZIONE DI UN NUMERO STABILITO DI COLONNE IN BASE AD UN NUMERO CONCORDATO DI ELIMINAZIONI	90.000
TF/5508-00	**APPLE PANIC (Ing.)** LABIRINTO DI SCALE - ALTA RISOLUZIONE GRAFICA AD ALTA VELOCITA	72.000
TF/5510-00	**ADVENTURES 1/2/3 (Ing.)** ADVENTURELAND - PIRATE'S ADVENTURE - MISSION IMPOSSIBLE	110.000
TF/5512-00	**ADVENTURES 4/5/6 (Ing.)** VODOO CASTLE - THE COUNT - STRANGE ODYSSEY	110.000
TF/5514-00	**ADVENTURES 7/8/9 (Ing.)** MYSTERY - FUN HOUSE PYRAMID OF DOOM GHOST TOWN	110.000
TF/5516-00	**FLIGHT SIMULATOR (Ing.)** UN REALISTICO SIMULATORE DI VOLO CON VISTA DAL CIELO E DALL'AEREO - ANCHE FASI DI COMBATTIMENTO	60.000
TF/5518-00	**COMPUCUBE (Ing.)** CREARE RIMESCOLARE - RISOLVERE IL CUBO MAGICO - TRIDIMENSIONALE	72.000
TF/5520-00	**DRAW POKER (Ing.)** IL MIGLIOR PROGRAMMA NEL SUO GENERE	72.000

Prezzi netti IVA esclusa

PERSONAL SOFTWARE

Raccolta di routine Basic

2

Mescolare

Questa routine (righe da 1150 a 1240) mescola in modo casuale i primi N numeri interi. Può servire per esempio a mescolare un mazzo di carte (N=40 o N=52). In termini matematici, la routine calcola una permutazione casuale dei primi N numeri.

Nelle righe da 1 a 5 vi è un esempio di utilizzo. Dapprima (riga 1) si deve dichiarare il valore di N, e dimensionare M con lo stesso valore. Quindi (riga 2) si entra alla prima riga della routine. Al ritorno (riga 3) si stampa la permutazione casuale ottenuta. La seconda volta che si chiama la routine (riga 4) e tutte le volte successive si può entrare direttamente alla riga 1180 evitando il riordinamento del vettore (che invece la prima volta è necessario perché deve essere creato).

Nella riga 1190 si è indicato con RND un numero casuale tra 0 e 1. L'istruzione J=INT(RND*I)+1 lo trasforma in numero casuale tra 1 e I. In alcuni Basic (per esempio TRS-80 e Sinclair ZX-80) ciò si può fare direttamente ponendo J=RND(I).

```
1 N=52: DIM M(52)
2 GOSUB 1150
3 FOR I=1 TO N: PRINT M(I),: NEXT I: PRINT: PRINT
4 GOSUB 1180
5 GOTO 3
1000 REM:   SCOPO
1010 REM:   DISORDINA CASUALMENTE I PRIMI N NUMERI (1,..N)
1020 REM:   IN ALTRE PAROLE, CALCOLA UNA PERMUTAZIONE
1030 REM:   CASUALE DEI PRIMI N NUMERI.
1040 REM:
1050 REM:   VARIABILI
1060 REM:   M(.)     CONTIENE I NUMERI MESCOLATI (IN USCITA
1070 REM:            E IN ENTRATA)
1080 REM:   N        DIMENSIONE DEL VETTORE M( )
1090 REM:
1100 REM:   VINCOLI
1110 REM:   N DEVE ESSERE UN INTERO
1120 REM:   M DEVE ESSERE DIMENSIONATO CON DIM M(N)
1130 REM:
1140 REM:   **********   INIZIO ROUTINE   **********
1150 FOR I=1 TO N
1160 M(I)=I
1170 NEXT I
1180 FOR I=N TO 2 STEP -1
1190 J=INT(RND*I)+1
1200 T=M(I)
1210 M(I)=M(J)
1220 M(J)=T
1230 NEXT I
1240 RETURN
```

L'organizzazione PERT

Una tecnica per la programmazione dei progetti

di *W. Douglas Maurer*

L'acronimo PERT sta per *Program Evaluation and Review Technique*, un metodo matematico usato da migliaia di programmatori di grandi e piccoli sistemi per risolvere uno dei problemi tipici dei manager del livello medio: come valutare la reciproca importanza dei lavori di cui sono responsabili.

Chiamiamo manager del livello medio una persona responsabile di un progetto composto da più lavori. Diversi manager del livello inferiore, preposti ai vari lavori, fanno riferimento al manager del livello medio. (D'altro canto, il manager di livello superiore si occupa prevalentemente della scelta dei progetti e della formulazione dei piani d'azione.) I compiti di base del manager del livello medio sono quelli di prevedere i possibili ostacoli e di completare il progetto nel tempo stabilito.

Un problema tipico

Per illustrare nel modo più chiaro il problema che i manager di livello medio devono affrontare, consideriamo il caso specifico in cui il progetto è la costruzione del quinto piano di un edificio. Il progetto inizia con la preparazione e la gittata del cemento armato. Per questo sono richiesti sei giorni, ma supponiamo che, per qualche motivo, ne siano stati impiegati sette. Per ora il progetto è in ritardo di un giorno.

A questo punto il manager esamina i vari lavori: mettere a piombo, isolare, e così via, e si accorge che, mentre quasi tutti richiedono dai tre ai cinque giorni, l'installazione del circuito elettrico ne occuperà sedici. Quindi assume qualche elettricista in più per effettuare il tutto in quattordici giorni. Ora il progetto è in vantaggio di un giorno sui programmi. Davvero? Dopo che i fili sono stati disposti, si passa a ricoprire ed intonacare, il che si può fare non appena finita l'isolazione del circuito. Questa richiede solo tre giorni, ma non può essere iniziata finché non sono terminati i controlli elettrici, che richiedono altri tre giorni. Ovviamente i controlli possono iniziare solo quando sono pronti i collegamenti e gli impianti, realizzati in cinque giorni... e avanti così. Alla fine il progetto è di nuovo in ritardo di un giorno.

Il problema di questo esempio (tolto, come gran parte del materiale per questo articolo, da *Fundamentals of Data Structures*, vedi bibliografia) è che la disposizione del circuito elettrico non è un'*operazione critica* (cioè un lavoro che causa lo slittamento dell'intero progetto se non è compiuto nel tempo previsto). Infatti in questo caso il manager avrebbe potuto assumere *meno* elettricisti, e consen-

La riproduzione di questo articolo è stata concessa da BYTE.
Traduzione di Flavio Santini

tire che il lavoro fosse effettuato in non più di ventotto giorni. Il denaro risparmiato potrebbe essere speso per l'assunzione di più operai per gli altri lavori che *sono* critici. Come può il manager determinare se un'operazione è critica o no? Ecco l'utilità del PERT.

L'analisi dei problemi col PERT

Il PERT può essere applicato in molti modi. Vediamo un semplice esempio. Per prima cosa ogni lavoro, o operazione, viene numerato, in modo che tutti i lavori possano essere eseguiti in ordine numerico. Per esempio, non possiamo pretendere che l'operazione numero 7 venga ultimata prima di iniziare la numero 4, perché in tal caso si poteva assegnare alla numero 4 un valore più alto di 7. Se ci sono *n* operazioni, esse vengono numerate da 1 a *n*.

Per assumere una notazione adatta al computer, costruiamo una matrice bidimensionale, chiamata B. Se vogliamo che, per ogni coppia I e J di lavori, I venga finito prima di iniziare J, poniamo B(I,J) = 1. Altrimenti B(I,J)=0. (Se non volessimo usare i doppi indici, potremmo adottare il seguente trucco: costruire un vettore A di N^2 elementi, dove n è il numero dei lavori, e riferirsi, invece che a B(I,J), ad A(K) se prima si è assegnato K=*n* per (I−1)+J. Così gli elementi da B(1,1) a B(1,*n*) vengono rappresentati con A(1),...,A(*n*); B(2,1),...,B(2,*n*) diventano A(*n*+1),...,A(2*n*); e così via.)

Inizializziamo la matrice a zero quindi inseriamo il valore 1 nei posti giusti, secondo la regola detta sopra. Costruiamo anche un vettore T, tale che T(I) è la quantità di tempo richiesta dall'operazione numero I. Se T(7)=5, significa che l'operazione numero 7 può essere realizzata in cinque giorni. (In effetti potrebbero essere cinque settimane, o cinque ore, purché in tutto il vettore T sia usata la stessa unità di misura.) Introduciamo tutti gli elementi di T, facendo variare I da 1 a *n*.

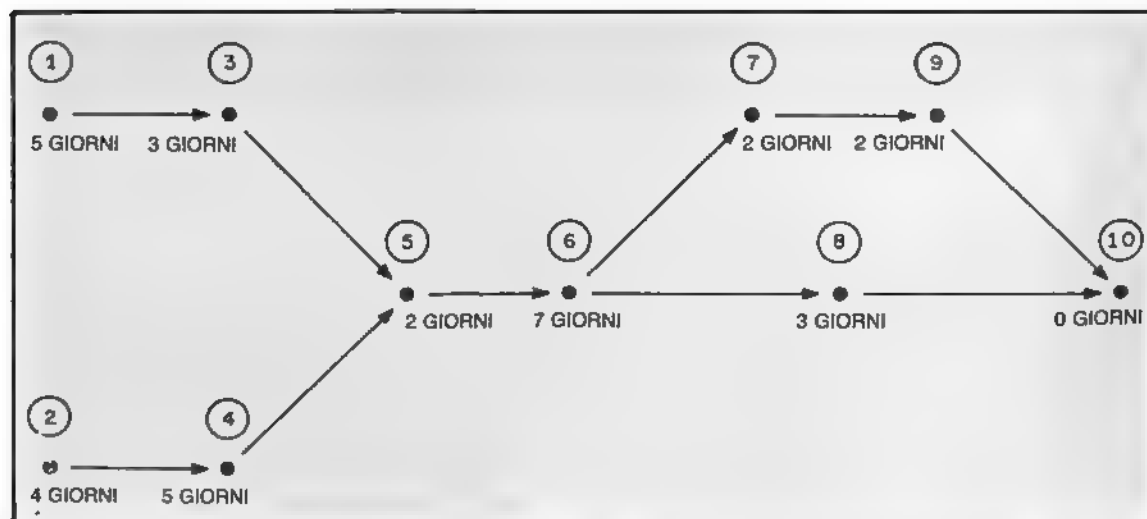



Fig. 1. *Un tipico progetto contenente dieci lavori, o operazioni. Ogni lavoro richiede un certo numero di giorni e può essere iniziato solo se quelli precedenti sono stati conclusi.*

Ora disponiamo di tutti i dati necessari, e possiamo dedicarci alla ricerca delle operazioni critiche. Prima dobbiamo creare un vettore, che chiameremo T1, in modo che T1(I) sia il tempo minimo d'inizio per il lavoro numero I. Se T1(5)=9, allora l'operazione numero 5 non può essere intrapresa prima del nono giorno del progetto. (D'ora in poi considereremo il giorno come unità di tempo).

La figura 1 illustra la ragione per cui T1(5) può essere uguale a 9. I numeri nei cerchietti corrispondono ai lavori, e sono collegati tra loro dalle freccie; tutte le operazioni raggiunte da freccie in entrata devono essere completate prima di iniziare l'operazione seguente. L'operazione numero 1 richiede 5 giorni e la numero 3 ne richiede 3. Se consideriamo solo la parte superiore del diagramma, possiamo desumere che il lavoro numero 5 può essere iniziato dopo otto giorni. Tuttavia, se prendiamo in esame il resto del diagramma, vediamo che, prima di passare alla numero 5, bisogna effettuare le operazioni numero 2, da quattro giorni, e numero 4, da cinque giorni. Perciò l'operazione numero 5 può, in effetti, essere iniziata solo dopo nove giorni.

(Spesso genera confusione il fatto che se un lavoro richiede tre giorni, e viene iniziato, diciamo, di lunedì, dovrebbe finire mercoledì. E mercoledì è due giorni, non tre, dopo lunedì. La soluzione a questo paradosso consiste nel considerare un giorno come un periodo di 24 ore. Se un lavoro è iniziato alle 8 di mattina di lunedì, e richiede tre giorni, diciamo che termina alle 8 di giovedì, mentre in realtà è finito alle 5 del pomeriggio di mercoledì.)

Nella figura 2, la freccia che congiunge le operazioni I e J corrisponde a B(I,J)=1. Così B(1,3)=1 e B(3,5)=1. Si potrebbe discutere se si debba porre B(1,5)=1 o no; dopo tutto, il lavoro numero 1 deve essere finito prima di iniziare il numero 5, ma solo in senso implicito. In questo caso, non ha importanza se B(1,5)=1.

In generale, le coppie ridondanti di operazioni possono essere date in input o tralasciate; il calcolo dell'operazione critica darà, comunque, lo stesso risultato.

Al variare di I da 1 a *n*, si effettua il calcolo del tempo minimo di inizio, T1(I). Ad ogni passo, consideriamo tutti i B(K,I) per K minore di I tali che B(K,I)=1. Se non deve terminare niente prima di poter iniziare l'operazione I, allora poniamo T1(I)=0, poiché in tal caso il lavoro I può cominciare al tempo zero. Nell'implementare il problema della figura 1, poniamo T1(1)=0 e T1(2)=0.

Se c'è *un* elemento B(K,I) che soddisfa la condizione sopra, allora aggiungiamo T1(K), il tempo minimo in cui si può iniziare il lavoro K, a T(K), il tempo richiesto dal lavoro K. Così, nella figura 1, per

calcolare T1(3), sommiamo T1(1) e T(1). Otteniamo che il lavoro 1 può partire al tempo zero, e richiede cinque giorni. Chiaramente, l'operazione 3 non può iniziare prima di cinque giorni dopo – quindi T1(3)=5. Allo stesso modo si calcola T1(4)=4.

> **Si deve supporre che l'ultima operazione non possa iniziare prima che le altre siano concluse.**

Se più di un elemento B(K,I) soddisfa la condizione, allora ripetiamo il calcolo più volte e scegliamo il risultato *maggiore*. Calcoliamo T1(5) basandoci sulla figura 1. Abbiamo:

T1(3)=5
T(3)=3
T1(3)+T(3)=8

e

T1(4)=4
T(4)=5
T1(4)+T(4)=9

Questo è il conto che abbiamo già fatto. La prima condizione dice che il lavoro numero 5 può iniziare solo dopo otto giorni; per la seconda invece solo dopo nove giorni. Ed è proprio questa la data che conta. In generale, potrebbe essere necessario considerare tre o più casi, e si dovrebbe assumere il risultato maggiore.

I risultati di T1(I), per tutti gli I, sono dati nella figura 2. Nei casi pratici, di solito, l'ultimo lavoro del progetto è la rifinitura, e non la si può effettuare prima di aver terminato tutto il resto. Nel prossimo calcolo, dobbiamo supporre che l'ultima operazione non può iniziare prima che tutte le altre siano concluse. Se non è così, consideriamo un lavoro fittizio, come il numero 10 nelle figure 1 e 2, che non richiede alcun tempo e chiude il progetto.

Se T(I)=J non è necessariamente vero che l'operazione numero I deve iniziare al tempo J. Consideriamo i lavori 6,7,9 e 10 della figura 2, e supponiamo che comincino tutti nei tempi stabiliti; cioè nell'undicesimo, diciottesimo, ventesimo e ventiduesimo giorno, rispettivamente. Ora prendiamo in esame l'operazione numero 8. Si è *programmato* di iniziarla nel diciottesimo giorno, ma potrebbe anche cominciare nel diciannovesimo (perché il lavoro richiede tre giorni, e quando lo si finisce si è arrivati al ventiduesimo – giorno finale del progetto).

Adesso siamo pronti ad iniziare un altro calcolo. Vogliamo calcolare un insieme di valori, chiamato T2(I), per tutti gli I da 1 a *n*, ma li calcoleremo in ordine inverso. Cioè, calcoleremo prima T2(*n*), poi T2(*n*−1) e così via, fino a T2(1).

Il valore T2(I) è il tempo *massimo* entro il quale si deve *finire* il lavoro numero I per non causare lo slittamento dell'intero progetto. Un attimo fa abbiamo visto che l'operazione numero 8 può essere effettuata nel diciottesimo o nel diciannovesimo giorno, e deve essere conclusa il ventunesimo o il ventiduesimo giorno. Quindi T2(8) sarà 22, perché il ventiduesimo giorno è il tempo massimo entro il quale può finire il lavoro numero 8.

Prima di passare al calcolo di T2(I), vediamo a cosa ci serve. Nel nostro esempio abbiamo T1(8)=18 e T2(8)=22. Cosa significa? Significa che l'operazione numero 8 *non può* iniziare prima del diciottesimo giorno, e deve finire entro il ventiduesimo. Quindi, questa operazione non può richiedere più di quattro giorni. Infatti abbiamo supposto che richieda tre giorni (T(8)=3) e può slittare di un giorno, ma non di più (altrimenti slitta l'intero programma). In questo caso l'operazione numero 8 *non* è critica. Se il programma dicesse che essa richiede quattro giorni – cioè se T(8) fosse uguale a 4 – *sarebbe* critica. Così appena calcoliamo T2(I) per tutti gli I, sappiamo immediatamente quali sono le operazioni critiche.

Per calcolare T2(I), consideriamo tutti i B(I,J), con J maggiore di I, tali che B(I,J)=1. Se non ci sono restrizioni (come, nelle nostre ipotesi, avviene per l'ultima operazione del progetto, cioè I=*n*) poniamo T2(I) uguale a T1(I)+T(I). Così l'ultimo lavoro deve iniziare al tempo T1(I) e richiede tempo T(I), e perciò deve terminare entro il tempo T1(I)+T(I) per far sì che l'intero progetto sia completato nel tempo minimo tenendo conto dei dati che abbiamo fornito per tutti i vari lavori.

Se c'è *un* B(I,J) che soddisfa la condizione sopra, allora sottraiamo T(J) (il tempo richiesto dal lavoro J) da T2(J), il tempo massimo in cui quel lavoro può terminare affinché il progetto rispetti i tempi del programma. Poiché i valori di T2(I) vengono calcolati in ordine inverso, possiamo supporre che T2(J) sia già stato calcolato. Nel

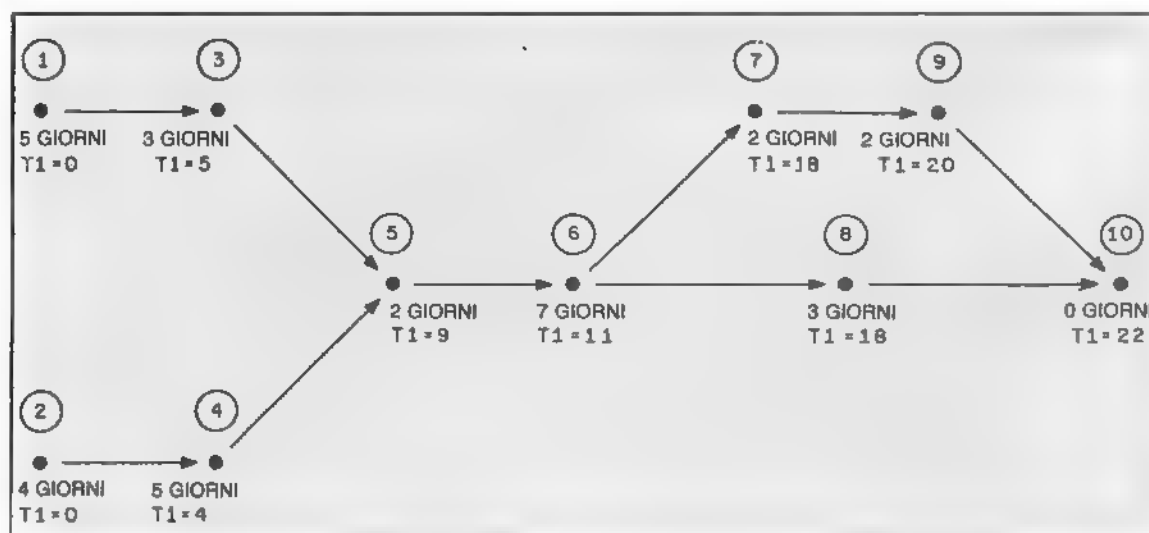



Fig. 2. *Per ogni lavoro nella figura 1, T1 (il minimo tempo in cui si può iniziare ogni operazione) può essere calcolato secondo i tempi ipotetici di realizzazione dei lavori precedenti.*

(1) 5 GIORNI T1=0 T2=6
(3) 3 GIORNI T1=5 T2=9
(2) 4 GIORNI T1=0 T2=4
(4) 5 GIORNI T1=4 T2=9
(5) 2 GIORNI T1=9 T2=11
(6) 7 GIORNI T1=11 T2=18
(7) 2 GIORNI T1=18 T2=20
(9) 2 GIORNI T1=20 T2=22
(8) 3 GIORNI T1=18 T2=22
(10) 0 GIORNI T1=22 T2=22

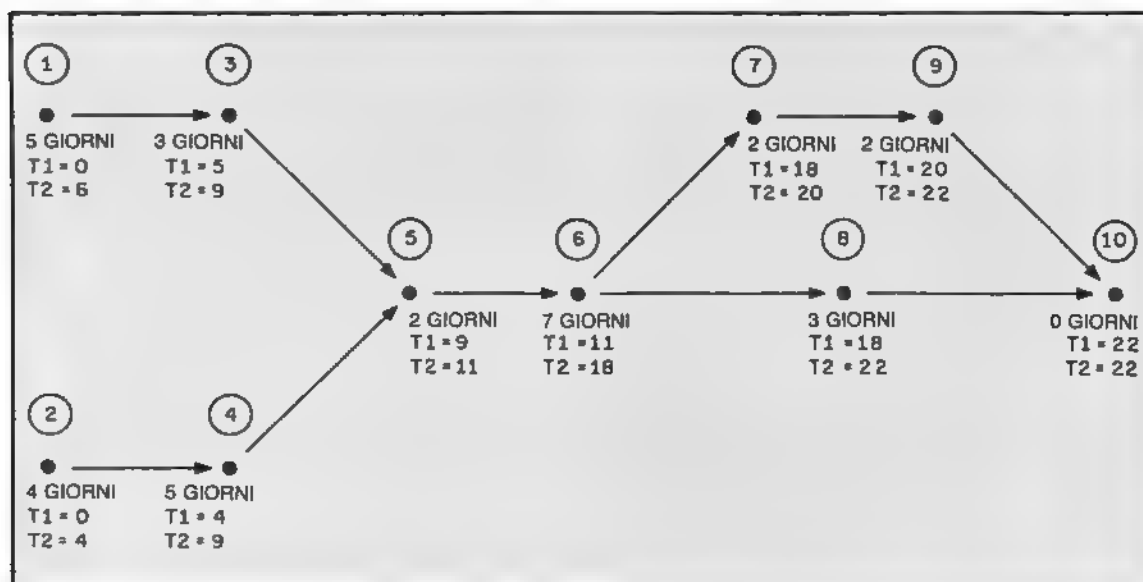



Fig. 3. *Per ogni lavoro nel progetto di figura 1 possiamo calcolare T2 (il tempo massimo entro il quale bisogna terminare quel lavoro per non far slittare l'intero progetto).*

progetto mostrato in figura 2, otteniamo il valore di T2(9) sottraendo T(11) da T2(11), perché l'operazione 10 non richiede alcun tempo, ed il risultato è 22. Troviamo T2(7) sottraendo T(9) da T2(9), ed il risultato è 20.

Si noti il significato di quest'ultimo risultato. Il nono lavoro richiede due giorni, e *deve* essere concluso entro il ventiduesimo giorno. Ciò significa che deve iniziare prima del ventesimo. Se guardiamo in figura 2, vediamo che quindi l'operazione numero 7 deve essere conclusa entro quel giorno. Analogamente, calcoliamo T2(8)=22.

Infine, se c'è *più di un* B(I,J) che soddisfa la condizione, allora effettuiamo più volte il calcolo e scegliamo il risultato *minore*. Ad esempio, se calcoliamo T2(6) nella figura 2, abbiamo:

T2(7)=20
T(7)=2
T2(7)−T(7)=18

e

T2(8)=22
T(8)=3
T2(8)−T(8)=19

Ciò significa che l'operazione numero 6 deve terminare entro il diciottesimo giorno, ed anche entro il diciannovesimo.Perciò, la data di scadenza per noi dev'essere il diciottesimo giorno. Possiamo notare da vari punti di vista che il calcolo di T2 è il contrario di quello di T1: procediamo a ritroso; consideriamo B(I,J) al posto di B(K,I); dobbiamo avere J maggiore di I, non più K minore di I; e, se ci sono più conti da fare, assumiamo il risultato minore, invece del maggiore.

Per accelerare un progetto, dobbiamo accelerare una attività che giace su tutti i percorsi critici.

I valori risultanti di T2(I), per ogni I, sono mostrati nella figura 3. Ora possiamo prendere in considerazione T1(I) e T2(I), per tutti gli I, e calcolare quali sono le operazioni critiche. Come abbiamo già notato, il lavoro numero I è critico se T2(I) − T1(I) = T(I); altrimenti non lo è. Le operazioni critiche nella figura 3 hanno i numeri 2, 4, 5, 6, 7, 9 e 10. Quelle non critiche sono 1, 3 e 8.

Ora abbiamo la risposta al problema del manager, in questo caso: i lavori 1, 3 e 8 non devono essere accelerati, perché non influiscono sul tempo d'esecuzione del progetto. D'altra parte, uno di questi tre lavori può slittare di un giorno senza modificare il tempo totale del progetto. (In effetti possono farlo le operazioni 1 e 8 oppure le operazioni 3 e 8, ma non sia la 1 che la 3.)

Le operazioni critiche giacciono tutte su un cammino che va dall'inizio alla fine del progetto. Lo si chiama *percorso critico*. In generale, in un progetto potrebbero esserci più percorsi critici. Se una operazione è critica, cioè se si trova un percorso critico, essa non può slittare nel tempo senza protrarre la durata complessiva del progetto. D'altro canto, non è detto che accelerando una particolare operazione si riduca il tempo totale: ciò avviene solo se essa giace su *tutti* i percorsi critici.

Considerazioni sulla codifica

Se il numero totale di operazioni è così grande che non riusciamo ad introdurre tutti gli elementi della matrice B(I,J) nell'area di memoria disponibile, possiamo usare il seguente trucco. Poiché ogni elemento B(I,J) è o 0 o 1 (la matrice B viene spesso chiamata *booleana*), possiamo inserire ogni elemento in un solo bit di una locazione di memoria. Su una macchina a 8 bit, lavorando in linguaggio assembler, rappresenteremo B(I,J) dividendo prima J per 8 ed ottenendo un quoziente K ed un resto L. Quindi potremmo memorizzare B(I,J) nell'L-esimo bit di B(I,K), e le dimensioni di B diventerebbero *n* per *n*/8 al posto di *n* per *n*.

Per ottenere questa rappresentazione, usiamo una tabella ausiliaria P, tale che l'elemento P(1) è il bit numero zero (da destra – cioè l'uno binario), l'elemento P(2) è il primo bit (cioè 2^1), P(3) è il secondo bit (cioè 2^2 o 4), e così via. Possiamo costruire quesha tabella ponendo P(1)=1 e quindi P(I+1)=2*P(I), per I che va da 1 a 7. Per ottenere l'L-esimo bit di X, usiamo l'operatore logico OR tra P(L+1) e X, e lo memorizziamo in X; per controllarlo, usiamo l'operatore AND di P(L+1) e X, e controlliamo il flag di zero. Con una macchina a 16 bit, facciamo la stessa analisi, sostituendo 16 a 8.

In Basic, anche su una macchina a 8 bit, ogni intero è solitamente memorizzato in 16 bit. Se nella versione di Basic in questione non sono disponibili le funzioni logiche AND e OR, oppure queste non sono presenti tra le funzioni di libreria, allora si può controllare l'L-esimo bit di X aggiungendolo a se stesso (cioè spostando di un bit a sinistra) N−1−L volte, e guardando se il risultato è negativo o no. Possiamo ottenere l'L'esimo bit di X *aggiungendo* P(L+1) ad X, purché si sappia che questo bit è libero (controllandolo come sopra).

Approfondimento

Troverete ulteriori dati sui percorsi critici e sulle operazioni critiche in *Fundamentals of Data Structures* di Ellis Horowitz e Sartaj Sahni. Gli autori descrivono due modelli grafici di un progetto – il modello AOV (*activity of vertex*), dove ogni vertice di un grafo come quello di figura 1 corrisponde ad un lavoro, ed il modello AOE (*activity on edge*), in cui i lavori sono rappresentati dagli archi che collegano i nodi. L'algoritmo del percorso critico che riportano gli autori si basa sul modello AOE, mentre quello che abbiamo descritto qui vale per il modello AOV. Gli autori discutono anche un algoritmo (chiamato *ordinamento topologico*) che serve per rinumerare tutte le operazioni, nel caso in cui la numerazione assunta non soddisfi la proprietà fondamentale di sviluppare tutte le operazioni del progetto secondo il numero d'ordine.

Bibliografia

HOROWITZ, ELLIS e SARTAJ SANHI *Fundamentals of Data Structures* Rockville MD: Computer Science Press, 1976.

PICCOLI ANNUNCI

Hardware

Vendo **Atari 400** + registratore Atari 410 (vero affare) L 800 000 vendo stampante Texas PC100C ancora in garanzia L 200 000
Luciano Medoni - Via Caboto. 5 - 30035 Mirano (VE) - Tel 041/431443

Vendo Sinclair **ZX-81** 16 Kbyte RAM alimentatore e documentazione L 400 000 non trattabili
Contatto possessori Texas TI-99/4A scambio esperienze e programmi
Salvatore Sbacchis - Via Don Giovanni Minzoni 2/E - 90143 Palermo - Tel 091/547670

Vendo 64 Kbyte **RAM Memotech** (G B C - MI) a L 300 000 causa prematura scomparsa dello ZX81 Telefonare ore 8-14 escluso la domenica
Rolando Magro - Via Sclavons. 302 - 33084 Cordenons (PN) - Tel 0434/26278

Vendo **VIC 20** + reg C2N + esp 16 Kbyte a L 800 000 intrattabili o cambio con fotocamera Reflex Olympus OM2N eventualmente Winder + tele 135 conguaglio
Roberto Orsucci - Via Pieve 96 - 55041 Camaiore (LU) - Tel 0584/68222

Vendo **Atari 400-800** cambio software originale americano utility e game su disco o cassetta Scrivere o telefonare ore pasti a
Marcello Guidotti - Via Cutilia 27 - 00183 Roma - Tel 06 778896

Vendo x ZX80-81 **espansione memoria** 4-16 32 Kbyte gen di caratteri programm Slow per ZX80 software su cassette o listati richiedere elenco pagamento anticipato o contrassegno
Alberto Bassi - Via M Bonavita - 47100 Forlì - Tel (0543) 24345

Vendo **DAI 48 Kbyte** + manuale italiano e inglese + processore aritmetico (anche a parte) + cavi + cassetta programmi Bit Shop 7 mesi di garanzia L 1 500 000 (P C) + 300 000 (AMD 9511)
Lorenzo Ambri Via Guido Guerra 12 50126 Firenze - Tel 680940

Software

Scambio **software** per VIC 20 - vendo ZX80 + 16 Kbyte RAM + alim + cavi + garanzia da convalidare + istr bilingue - assemblato in fabbrica - Vinto ad un concorso L 300 000 usato 10 giorni
Amos Aimi - Via Zanella 11 - 43015 Noceto (PR

Cambio/vendo **programmi** per VIC 20 vendo cassetta con 5 giochi per VIC 20 L 15 000 Per VIC 20 vendo cass con equo canone + isolamento termico L 15 000
Maurizio Mel one - Via Sabbionara 9 - 36061 Bassano del Grappa (VI) - Tel 20015

Atari **Club** per scambio programmi più di cento programmi animazione grafica musica games uno più favoloso dell altro scrivetemi per saperne di più W gli ataristi!!
Marco Rosati - Via Giovanni XXIII - 10040 Caprie (TO)

Cambio **programmi** per Apple II e TRS 80 gestionali utilities giochi vari Inviare lista, invierò la mia
Giovanni Carrella - Via Felline 11 C - 84100 Salerno - Tel 089 354854

Desidererei comprare **software** impianti termici (specificare caratteristiche e *prezzo*) riscaldamento analisi sismica spaziale completo di verifica (prezzo e caratteristiche
Edoardo Cardone Via S Matteo 36 71016 San Severo - Tel 27447

Cambio **programmi** di tutti i generi per Apple II o vendo a prezzi di sole spese spedizione e di supporto magnetico Inviare lista a cui risponderò con la mia
Adriano Gugliucci - Viviani Via Massari 100 ‑ 10148 Torino - Tel 011 2201756

Vendo **cassetta** programmi gioco per VIC 20 e altri programmi chiedere lista a
Claudio Marchiondelli Via Libertà 3 33010 Raspano (UD)

Cambio vendo **software** per Apple I
Marco Montresor P zza C Alberto 25 37067 Valeggio S M (Verona)

PICCOLI ANNUNCI PERSONAL SOFTWARE

Sei un lettore di PERSONAL-SOFTWARE e vuoi entrare in contatto con tutti gli altri lettori per comperare, cambiare o vendere software? Spedisci questo tagliando a Gruppo Editoriale Jackson - Via Rosellini, 12 - 20124 Milano

Nome .... Cognome ....

Via .... C.A.P ....

Città .... Tel. ....

P.G.P. SISTEMA

via soperga, 36 - 20127 MILANO - 02/2842850-2842860

PROPONE

MZ - 80 B	PC - 3201	MZ - 80 A

La famiglia di personal computer più completa attualmente sul mercato per imprenditori, professionisti, tecnici, amministratori.
Vi aspettiamo per fornirvi la più ampia documentazione e le più complete dimostrazioni.

SHARP COMPUTERS

I NOBEL DELL'INFORMATICA

PIEMONTE - GENERAL COMPUTERS - Torino - 011/835156 - COMPDATA - Ivrea - 0125/49069 - OLIVIERI & GOVERNA - Alessandria - 0131/442646 - **LIGURIA** - REM KARD ITALIA - Genova 010/884971 - TECNOSYSTEM - Sanremo - 0184/884794/5 - **LOMBARDIA** SHARCO (di MIGLIORI ROBERTO) - Gavirate - 0332/745526 - ADEL - Brescia - 030/221674 - Peschiera Borromeo - 02/5473023 - GAME - Treviglio - 0363/40803 - ENNE COMPUTER - Portichetto di Luisago - 031/920136 - C.E.E. - Vigevano - 0381/81555 - DATA STUDIO (di SERGIO CAVENAGHI) - Burago di Molgora - 039/663736 - LINEA UFFICIO (di ANNUNZIATA ELIO) - Cremona - 0372/24364 - P.G.P. SISTEMA - Milano - 02/2842860 - COMPUTER HOUSE - Monza - 039/362618 **TRE VENEZIE** - SIGMA SYSTEM - Udine - 0432/26992 - INTERSOUND (di COPPETTI FRANCO) - Brunico - 0474/21282 - COMMERCIALE SISTEMI Thiene - 0445/368824 - MINI SYSTEM - Bolzano - 0471/32270 - PENTA - Preganziol - 0422/938535 - PINARELLO - Padova - 049/754830 - SYSTEM COPY - 35100 Padova - 049/44982 - PINO ANDREA - Cerea - 0442/82790 - TECNOSYSTEM - Vicenza - 0444/31152 - **EMILIA ROMAGNA - MARCHE - ABRUZZO** - ADRIATICA COMPUTER - Senigallia - 071/62516 - GIMAR SISTEMI - Silvi Marina - 085/932739 - ZANICHELLI GIORGIO - Reggio Emilia - 0522/90250 - M.R.P. TECNOSISTEMI - 40069 Zola Predosa - 051/751662 - RODAN & C. - Civitanova Marche - 0733/770386 - ROGANTI F.LLI - Montecassiano - 0733/59231 **TOSCANA** - ELECON - Piombino - 0565/33112 - MNEMO COMPUTERS - Firenze - 055/4378652 - TECNOCOPY - Firenze - 055/352801 - **LAZIO** - EUROCOM - Roma - 06/7574487 - TECNOMEC - Roma - 06/484998 - **CAMPANIA - PUGLIE - CALABRIA** - GENERAL COMPUTERS - Torre del Greco - 081/8815124 - L. & L. COMPUTERS - Bari - 080/410167 - COMPUTER SUD - Lecce - 0832/42413 - ATLANTIC - Reggio Calabria - 0965/44671 - G.M. MARASCIO COMPUTERLINE - Montauro - 0967/48207 - **SICILIA - SARDEGNA** - SIFI DATA MANAGEMENT - Catania - 095/438178 - A.E.P. COMPUTERS SYSTEM - Sassari - 079/276364 - VIMAR - S. Agata di Militello - 0941/702771.

SHARP

DESIDERO

☐ RICEVERE UNA DOCUMENTAZIONE SULLE SOLUZIONI CON I COMPUTERS SHARP

☐ DISCUTERE IL MIO PROBLEMA SPECIFICO CON UN VOSTRO INCARICATO

NOME

SOCIETÀ — POSIZIONE

INDIRIZZO

CITTÀ — TEL

La programmazione discendente

Nino Tonolli

L'arte di programmare con stile ed efficacia. Questa rubrica conterrà raccomandazioni su come scrivere i programmi: non solo consigli tecnici, ma soprattutto consigli di stile, sul modo di affrontare i problemi e trasformarli in algoritmi, in buoni algoritmi e poi in buoni programmi. Questo mese proseguiamo con altri "proverbi di Arsac", le raccomandazioni del professore francese per acquisire "buone abitudini di programmazione".

6. Un solo ciclo alla volta

Separate le difficoltà. Una sola alla volta (ce n'è a sufficienza). L'esempio del mese scorso mostra come fare. È vero che servirà un ciclo per calcolare il valore del polinomio, ma ci si è solo occupati del ciclo più esterno. Ancora non si sa se servirà un ciclo per le altre azioni che si sono dette. Ciò sarà più chiaro con uno studio ulteriore.

Il meccanismo dei cicli non è complicato, ma i rischi di errore sono numerosi: errata inizializzazione, errato computo dei passi, errato test di uscita... Allora, non mescolate le difficoltà: una sola alla volta. È sempre possibile, grazie alla programmazione discendente.

7. Fondate tutti i cicli su una affermazione dello stato delle variabili

Un ciclo è la ripetizione di una azione. È la forma che prende, in programmazione, il *ragionamento per ricorrenza*. Operate dunque come avete l'abitudine di fare: al centro di tutto vi è l'affermazione: suppongo di essere arrivato ad uno stato tale che... Si guarda se per caso non sia lo stato finale. Se sì, si esce dal ciclo. Altrimenti, si cerca come ritrovare lo stesso stato, più vicino alla soluzione.

Invece di restare nel vago, prendiamo il calcolo del valore y del polinomio in x

$$y = \sum_{p=0}^{n} a(p)x^p$$

Prima constatazione: questo calcolo non si può fare in un solo passo. Bisogna dunque spezzare la somma.

Supponiamo che la si spezzi verso il basso. Non ho ancora calcolato y, ma solamente

$$z = \sum_{p=i}^{n} a(p)x^{p-i}$$

Ci vuole dell'immaginazione per fare questa ipotesi. Ma l'immaginazione è necessaria in informatica come in tutte le scienze. (Fortunatamente, del resto. La macchina ha già molta influenza sull'uomo. Cosa ci resterà se ci si leva l'immaginazione...)

Domanda: z è il risultato cercato? Risposta: sì, se $i = 0$. Da cui un pezzo di programma, ancora in programmazione discendente:

se $i = 0$ allora fine

Altrimenti, bisogna estendere la somma verso $i = 0$, ed il modo più semplice è aggiungere $i-1$. Si ottiene

$$z' = \sum_{p=i-1}^{n} a(p)x^{p-i-1} = \sum_{p=i}^{n} a(p)x^{p-i+1} + a(i-1)$$

ordinando questa somma

$$z' = (\sum_{p=i}^{n} a(p)x^{p-i})x + a(i-1) = zx+a(i-1)$$

Possiamo ora proseguire con la costruzione del programma. Metterò tra virgolette dei commenti sullo stato delle variabili.

"$z = \sum_{p=i}^{n} a(p)x^{p-i}$" se $i = 0$ allora fine "$z = y$"

$z \leftarrow z*x+a[i-1]$ "$z = \sum_{p=i-1}^{n} a(p)x^{p-i+1}$"

Non abbiamo ritrovato l'ipotesi di ricorrenza, i è stato cambiato in $i-1$. È sufficiente cambiare $i-1$ in i per ritrovarci al punto di partenza:

H: "$z = \sum_{p=i}^{n} a(p)x^{p-i}$" se i=0 allora fine

$z \leftarrow z*x+a[i-1]$ "$z = \sum_{p=i-1}^{n} a(p)x^{p-i+1}$"

$i \leftarrow i-1$ "$z = \sum_{p=i}^{n} a(p)\, x^{p-i}$"

ELETTRONICA INTEGRATA DIGITALE

di Erbert Taub e Donald Schilling'

Pag. 740
Formato 16,5x23
Cod. 204A

L. 34.500
(Abb. L. 31 050)

Non esiste, in lingua italiana, un libro di testo così. Chiaro, completo, moderno, ma anche rigoroso e didattico. Sono alcuni tra gli aggettivi che costituiscono la prerogativa di questo volume. Per capire l'elettronica digitale bisogna avere delle solide conoscenze sui dispositivi a semiconduttore, soprattutto usati in circuiti di commutazione.
E malgrado quest'analisi richieda una notevole complessità matematica, introducendo alcune semplificazioni è possibile mantenere la trattazione ugualmente rigorosa e ottenere approssimazioni pienamente accettabili Come trascurare poi gli amplificatori operazionali, che, se a rigore non rientrerebbero nella materia, però trovano larga applicazione in sistemi completamente digitali E poi i circuiti integrati, finalmente spiegati e analizzati in tutti i loro aspetti. Dalla vecchia logica resistore-transistor (RTL), funzionale nella sua semplicità all'esemplificazione degli aspetti fondamentali, a quella a simmetria completamente (CMOS)
Questo, però, dopo aver studiato un capitolo che, pur non richiedendo alcuna conoscenza preliminare, va a fondo dei concetti di variabile logiche, di algebra di Boole, di analisi di circuiti logici E ancora. Via via nei vari capitoli: i flip-flop, i registri, e i contatori (sia sincroni che asincroni), i circuiti logici atti ad eseguire operazioni matematiche, le memorie a semiconduttore (RAM, ROM, EPROM, ...), l'interfacciamento tra segnali analogici e digitali (multiplexer, circuiti sample and hold, convertitori d/a e a/d), i temporizzatori. Tutto con oltre 400 problemi, dai più semplici ai più sofisticati, in cui vengono presentati i circuiti tipici che si trovano nella pratica
Un testo quindi non solo per gli specialisti e per gli studenti universitari, ma che si adatta magnificamente agli Istituti Tecnici.
Un testo che, speriamo per gli studenti, la scuola non debba scoprire tra alcuni anni

SOMMARIO
Dispositivi Elettronici fondamentali; AMplificatori Operazionali e Comparatori, Circuiti Logici, Logica Resistore-Transistore e Logica ad Iniezione Integrata; Logica Diodo-Transistore, Logica Transistore-Transistore, Logica ad Accoppiamento di Emettitore, Porte MOS, I Flip-Flop; Registri e Contatori; Operazioni Aritmetiche; Memorie a Semiconduttore; Interruttori Analogici, Conversione Analogico-Digitale; Circuiti di Temporizzazione; Linee di Trasmissione, Problemi; Alcuni Esempi di Specifiche

GRUPPO EDITORIALE JACKSON
Divisione Libri

Per ordinare il volume utilizzare l'apposito tagliando inserito in fondo alla rivista.

8. Utilizzate correttamente le istruzioni GO TO

Non si va ad una istruzione per fare qualcosa, ma perché essa corrisponde ad una situazione già identificata, o per lo meno da identificare. Non pensate in termini di "quale azione devo ora intraprendere". Questa è la morte del programmatore, la miglior ricetta per fallire. La domanda da porre è "a che punto sono?". Se operate così non avrete alcun problema con questa istruzione.

Troverete delle diramazioni per due ragioni:

— mi trovo in uno stato già incontrato: vado là dove è stato trattato. Questo è il salto di ciclo. Nell'esempio precedente, ci si è arrestati proprio qui. Si era ritrovato lo stato H: quindi

  H: "$z = \sum_{p=i}^{n} a(p)x^{p}$"   se $i = 0$ allora fine
  $z \leftarrow z^*x+a[i-1]$; $i \leftarrow i-1$; vai a H

— mi trovo in uno stato che non desidero per il momento trattare. Lo rinvio a più tardi, in un punto dove lo tratterò. Si tratta del salto in avanti, risultante in generale da una selezione (istruzione SE). Così, nell'esempio del polinomio, l'azione "finito" corrisponde allo stato z = y, risultato cercato. Lo rinvio in avanti:

  H: "$z = \sum_{p=i}^{n} a(p)x^{p-i}$"
  se $i = 0$ allora "$z = \sum_{p=0}^{n} a(p)x^{p}$" vai a B;
  $z \leftarrow z^*x+a[i-1]$; $i \leftarrow i-1$; vai a H
  B: $y \leftarrow z$

9. Scrivete l'inizializzazione dopo il ciclo

Avete fondato il vostro ciclo su di una affermazione: suppongo di poter raggiungere uno stato tale che... Avete determinato il caso in cui si tratta dello stato finale, altrimenti avete trovato un modo di avvicinare la soluzione ritrovando l'ipotesi di partenza.

Ora cercate come ottenere questa ipotesi alla partenza: si tratta dell'inizializzazione del ciclo. Se avete lavorato in modo appropriato non avrete scelta, né bisogno di immaginazione. Qui per esempio, come scegliere i per ottenere, senza calcolo

$$z = \sum_{p=i}^{n} a(p)x^{p-i}$$

Se si prende $i = n$, la sommatoria comporta un solo termine, $a(n)x^{n-n} = a(n)$

Per partire, bisogna dunque scrivere

$i \leftarrow n$; $z \leftarrow a[n]$
H: se $i = 0$ vai a B
$z \leftarrow z^*x+a[i-1]$; $i \leftarrow i-1$; vai a H
B: $y \leftarrow z$

Una nuova generazione di italiani

PRAKTI 80 - A. VALERI

General Processor Sistema 4

GPS4 è il nome della nuova famiglia di elaboratori General Processor: elaboratori perfetti, nati dalla esperienza della prima azienda italiana costruttrice di piccoli computer.
I GPS4 sono tutti italiani: italiani nel progetto, italiani nella costruzione, italiani nel design, elegante ed essenziale come quello di un'auto sportiva di gran classe. Hanno una tastiera italiana, separata, davanti alla quale ogni dattilografa si trova subito a suo agio perché la Z, la W e la M sono al loro posto e perché, come in una calcolatrice, ci sono i tasti doppio e triplo zero.
E sono italiani anche nella assistenza.
Con i loro 128K RAM minimi (estendibili a oltre 200), due terminali collegabili e con una ineguagliabile biblioteca di software di base ed applicativo, i GPS4 rappresentano lo "status of the art" della moderna miniinformatica, per la quale rappresentano e rappresenteranno negli anni futuri un importante punto di riferimento.
Una raccomandazione: non fatevi influenzare dallo styling: i GPS4 non sono semplicemente i più belli; sono semplicemente i migliori.

Alcuni OEM General Processor

Milano: PGE: 02/28.22.225 - Como e Varese: SIAEMME: 0331/67.96.75 - Alessandria: CID: 0131/34.44.18 - Modena: Data: 059/68.80.90 - Bologna: Computer Systems: 051/79.94.21 - Pistoia: CEIA: 0572/51.611 - Firenze: R2 Data: 055/41.11.42 - Firenze: Aeffe: 055/75.27.89 - Prato: Gerva: 0574/59.26.94 - S. Croce/Arno (PI): Dainelli: 0571/31.805 -Arezzo: Tecem: 0575/28.848 - Arezzo: Etruria Sistemi: 0575/35.59.71 -Livorno: CEDOS: 0586/25.395 - Siena: Tecnocomputer: 0577/74.03.34 -Roma: General Computer: 06/52.84.032 - Latina: Contax: 0771/22.503 -Napoli: CompuSystems: 081/46.36.02 - Napoli: Tecnodata: 081/24.21.66 - Calabria: Tripodi: 0984/99.21.42 - Spagna (Madrid)/Vimesa: 690.20.29

GENERAL PROCESSOR s.r.l.- elaboratori italiani - Firenze
Tel. 055/43.55.27 - 43.763.88 - Tlx 571034 GENPRO I

Immagini digitali di mezzetinte

Come riprodurre una immagine in bianco e nero con toni sfumati?

di *J.S. Browning*

Molti programmatori dilettanti interessati all'elaborazione di immagini digitali vengono frustrati dai limitati intervalli di intensità disponibili sui dispositivi di output sui quali devono visualizzare le proprie immagini. Le tecniche per risolvere questo problema sono dette "metodi delle mezzetinte".

In origine i metodi delle mezzetinte vennero sviluppati per la stampa di fotografie in bianco e nero su carta, usando sfumature di inchiostro nero. L'obiettivo è di visualizzare strutture di bianco e nero che diano l'impressione di intensità intermedie.

Fotograficamente, ciò si realizza usando un retino da mezzatinta. Esempi di fotografie a mezzatinta si trovano nella maggior parte dei giornali e delle riviste.

Un problema simile si presenta quando si desidera visualizzare una immagine digitale su un dispositivo di output "a due livelli".

In questo caso si fa qualcosa di simile alla mezzatinta analogica. Si usa un computer per generare delle strutture che possano essere visualizzate sul dispositivo in modo da dare l'impressione di intensità intermedie.

In questo articolo, presento diversi metodi di mezzatinta, e darò esempi di applicazione dei metodi ad una immagine. Nella discussione che segue, userò queste notazioni:

G(X,Y) = livello di grigio del pixel (elemento di immagine) X,Y
T = livello di grigio di soglia

Soglia semplice

Usando il metodo della soglia semplice, se G(X,Y) è minore di T, (X,Y) viene visualizzato nero. Altrimenti (X,Y) viene visualizzato bianco. La figura 1 mostra un esempio di questo metodo.

Il metodo della soglia semplice non mostra parecchi dettagli presenti nell'immagine perché la decisione di visualizzare o bianco o nero introduce degli errori.

Soglia a spirale

Con il metodo della soglia a spirale, definiamo una matrice soglia a spirale M,N come segue:

T(I,I) ha un certo valore
T(I,I+1) è minore di T(I,I)
T(I−1,I+1) è minore di T(I,I+1)
T(I−1,I) è minore di T(I−1,I+1)
T(I−1,I−1) è minore di T(I−1,I)
T(I,I−1) è minore di T(I−1,I−1)

e così via. T(I,I) è il centro della matrice T.

L'immagine viene ripartita in sottoimmagini M per N, ed ogni componente della matrice sottoimmagine viene confrontato con il corrispondente componente della matrice soglia a spirale. Se G(I,J) è minore di T(I,J), (I,J) viene visualizzato nero. Altrimenti (I,J) viene

La riproduzione di questo articolo è stata concessa da Creative Computing.
Traduzione di Renzo Pevarolo

Questo articolo didattico descrive sette metodi per produrre immagini digitali di mezzetinte. I programmi esemplificativi sono in Fortran. Avremo potuto tradurli facilmente in Basic, ma abbiamo scelto di lasciarli in Fortran per permettere un secondo livello di apprendimento nella lettura dell'articolo. Il Fortran ha più somiglianze con il Basic di questo comunemente non si pensi. Per esempio il seguente ciclo DO Fortran può essere facilmente tradotto in un ciclo FOR Basic.

Fortran	Basic
DO 20 J=1021, 1024 G(J)=O 20 CONTINUE	10 FOR J=1021 TO 1024 15 G(J)=O 20 NEXT J

Il Fortran permette che i limiti del ciclo DO siano definiti da variabili semplici o da semplici espressioni. Il Basic talvolta non permette espressioni.

Fortran	Basic
DO 20 M=K−1,K+1	10 K1=K−1: K2=K+1 20 FOR M=K1 TO K2

L'istruzione IF logica nel Fortran è simile a quella del Basic.

Fortran	Basic
IF(G(M,N).LT.T(M,N)) G(M,N)=1	10 IF G(M,N) < T(M,N) THEN G(M,N)=1

Un'istruzione DATA Fortran può essere tradotta in Basic con una coppia READ/DATA o con istruzioni di assegnamento LET.

Fortran	Basic
INTEGER T(5) DATA T/7,9,11,13,15/	10 DIM T(5) 15 FOR I=1 TO 5 20 READ T(I) 25 NEXT I 30 DATA 7,9,11,13,15 oppure 10 T(1)=7: T(2)=9: T(3)=11 15 T(4)=13: T(5)=15

Per le variabili a due dimensioni, in una istruzione DATA si può inserire un ciclo di DO implicito; per esempio:

Fortran	Basic
INTEGER P(2,5) DATA (P(1,I),I=1,5)/4,5,6,7,8 DATA (P(2,I),I=1,5)/13,14,15,16,17	10 DIM P(2,5) 15 FOR I=1 TO 5 20 READ P(1,I),P(2,I) 25 NEXT I 30 DATA 4,13,5,14,6,15,7,16,8,17

Questi suggerimenti permettono di convertire le routine dell'articolo in Basic.

visualizzato bianco (figura 2). In Fortran abbiamo per esempio:

```
      SUBROUTINE SPIRAL
      INTEGER T(5,5),G
      COMMON G(1024,1024)
      DATA (T(1,I),I=1,5)/90,100,110,120,130
      DATA (T(2,I),I=1,5)/80,210,220,230,140
      DATA (T(3,I),I=1,5)/70,200,250,240,150/
      DATA (T(4,I),I=1,5)/60,190,180,170,160/
      DATA (T(5,I),I=1,5)/50,40,30,20,10/
      DO 10 J=1,204
      L=J*5-2
      DO 10 I=1,204
      K=I*5 2
      DO 10 N=L-2,L+2
      DO 10 M=K-2,K+2
C     PIXEL NERO
      IF (G(M,N).LT.T(M-K+3,N-L+3)) G(M,N)=1
C     PIXEL BIANCO
      IF (G(M,N).GE.T(M-K+3,N-L+3)) G(M,N)=0
   10 CONTINUE
C     AZZERA I PIXEL NON USATI
      DO 20 J 1021,1024
      DO 20 I=1,1024
      G(I,J)=0
   20 CONTINUE
      DO 30 J=1,1024
      DO 30 I=1021,1024
      G(I,J)=0
   30 CONTINUE
      RETURN
      END
```

Il metodo della soglia a spirale dipende dalla continuità dell'immagine, cioè G(X,Y) adiacenti sono quasi uguali. Il metodo produce un "punto" le cui dimensioni dipendono da G(X,Y). Inoltre, il metodo è abbastanza trasparente ai cambi repentini in G(X,Y) dovuti ad uno spigolo.

Soglia a rettangolo

Per utilizzare il metodo della soglia a rettangolo, definiamo un vettore soglia lungo K dove i vari T sono distribuiti casualmente ed hanno pochi fattori comuni. L'immagine è suddivisa in sottoimmagini M,N con MN=K. Ogni componente della sottoimmagine è confrontato con un componente del vettore soglia. La corrispondenza tra M, N e K è definita evitando una trama eccessiva.

Per esempio:

```
      SUBROUTINE RETTANG
      INTEGER T(25),G
      COMMON G(1024,1024)
      DATA T/40,80,140,120,20,60,170,210,
     .   190,100,150,240,250,230,160,
     .90,200,220,180,150,10,110,130,70,30/
      IB=1
      IT=IB
      DO 10 J=1,1024
      DO 10 I=1,1024
C     PIXEL NERO
      IF (G(I,J).LT.T(IT)) G(I,J)=1
C     PIXEL BIANCO
      IF (G(I,J).GE.T(IT)) G(I,J)=0
C     SCELTA DI T
      IB=IB+1
      IF (IB.GT.25) IB=1
      IT=IB
   10 CONTINUE
      RETURN
```

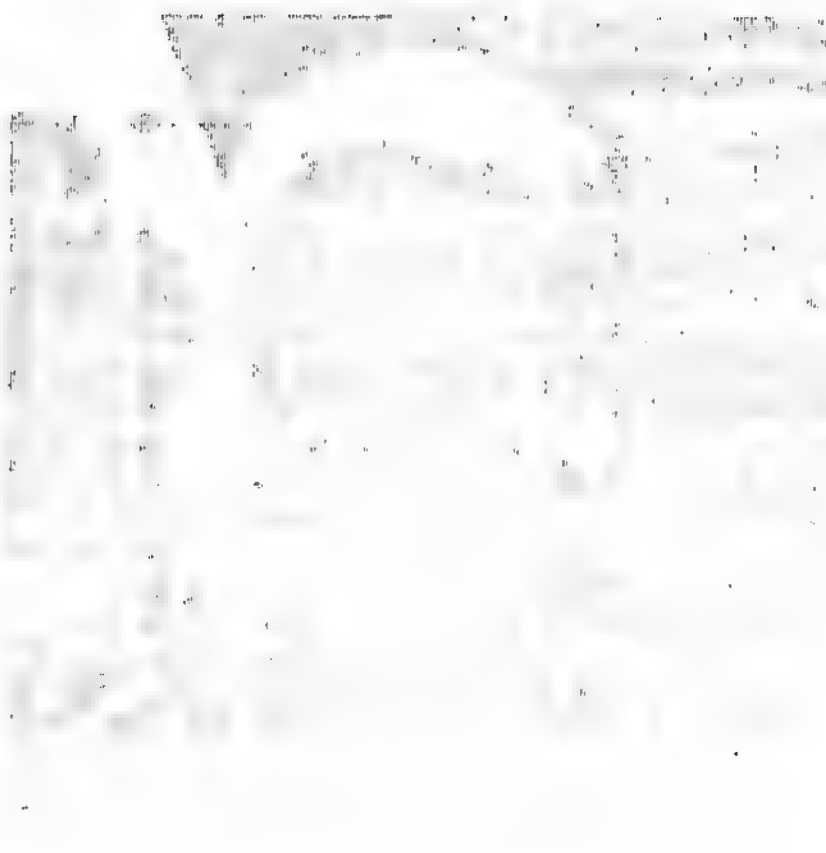



Fig. 1. *Simple Threshold Method.*

Fig. 2. *Spiral Threshold Method.*

Come il metodo della soglia a spirale, il metodo della soglia rettangolare è trasparente ai cambi repentini in G(X,Y) dovuti a spigoli. Vedi la figura 3.

Modulazione casuale

Nella modulazione casuale, un generatore di numeri pseudocasuali produce un segnale usato per modulare G(XY).

Se G(X,Y)+T*(1−2*R) è minore di T, (X,Y) viene visualizzato nero. Altrimenti, (X,Y) viene visualizzato bianco. L'output del generatore di numeri casuali è R ed è compreso tra 0 e 1 con media 0.5. Nell'esempio (figura 4), il generatore di numeri casuali è S=FRAC (S+π)**5, e il seme è 0.192743568.

Il metodo della modulazione casuale introduce un rumore "sale e pepe" nella visualizzazione, che può essere inaccettabile.

Modulazione sinusoidale

Nella modulazione sinusoidale viene usata la funzione SIN(A*X) *SIN(B*Y) per modulare G(X,Y).

Se G(X,Y)+T*SIN(A,X)*SIN (B,Y) è minore di T, (X,Y) viene visualizzato nero. Altrimenti (X,Y) viene visualizzato bianco.

Il metodo della modulazione sinusoidale è simile alla tecnica della soglia a spirale. Le costanti di modulazione, A e B, vengono scelte in modo da evitare una trama eccessiva. Nell'esempio (figura 5) le costanti di modulazione sono π/3.

Distribuzione degli errori

Il metodo di distribuzione degli errori registra gli errori che vengono introdotti dalla soglia semplice e li distribuisce ai pixel adiacenti.

```
      SUBROUTINE DISTERR
      INTEGER G,T
      COMMON G(1024,1024),T
      DO 10 I=1,1024
      IP=I+1
C
      IF (IP.GT.1024) IP=1
      DO 10 I=1,1024
      J=1024-K+1
      JM=J-1
C     DISTRIBUZIONE ERRORE
      IF (JM.LT.1) JM=1024
C     NERO=0
      IF (G(I,J).LT.T) ERROR=G(I,J)
C     BIANCO=255
      IF (G(I,J).GE.T) ERROR=G(I,J)-255
C     DISTRIBUZIONE ERRORE A DESTRA
      G(IP,J)=G(IP,J)+3.*ERROR/8.
C     DISTRIBUZIONE ERRORE SOTTO
      G(I,JM)=G(I,JM)+3.*ERROR/8.
C     DISTRIBUZIONE ERRORE IN DIAGONALE
      G(IP,JM)=G(IP,JM)+ERROR/4.
   10 CONTINUE
      RETURN
      END
```

Man mano che l'errore si propaga, nell'immagine vengono introdotte strane strutture.

Nell'esempio (figura 6), l'errore è distribuito come seguente: 3/8 alla destra di (X,Y), 3/8 sopra (X,Y) e 1/4 lungo la diagonale.

Struttura

La tecnica della struttura usa K+1 matrici di K×K punti per visualizzare K+1 livelli di grigio. Per

```
      SUBROUTINE STRUTTURA
      INTEGER P(10,9),G,T
      COMMON G(1024,1024)
      DATA (P(1,I),I=1,9)/9*0/
      DATA (P(2,I),I=1,9)/4*0,1,4*0/
      DATA (P(3,I),I=1,9)/4*0,2*1,3*0/
      DATA (P(4,I),I=1,9)/0,1,2*0,2*1,3*0/
      DATA (P(5,I),I=1,9)/0,1,0,3*1,3*0/
      DATA (P(6,I),I=1,9)/0,1,0,3*1,0,1,0/
      DATA (P(7,I),I=1,9)/0,1,0,5*1,0/
      DATA (P(8,I),I=1,9)/0,7*1,0/
      DATA (P(9,I),I=1,9)/8*1,0/
      DATA (P(10,I),I=1,9)/9*1/
      DO 10 I=1,341
      K=I*3-1
      DO 10 J=1,341
      L=J*3-1
      T=G(K,L)/26
      DO 10 M=K-1,K+1
      DO 10 N=L-1,L+1
   10 G(M,N)=P(10-T,M-K+2+3*(N-L+1))
C     AZZERAMENTO PIXEL NON USATI
      J=1024
      DO 20 I=1,1024
      P(I,J)=0
20    CONTINUE
      I=1024
      DO 30 J=1,1024
      P(I,J)=0
   30 CONTINUE
      RETURN
      END
```

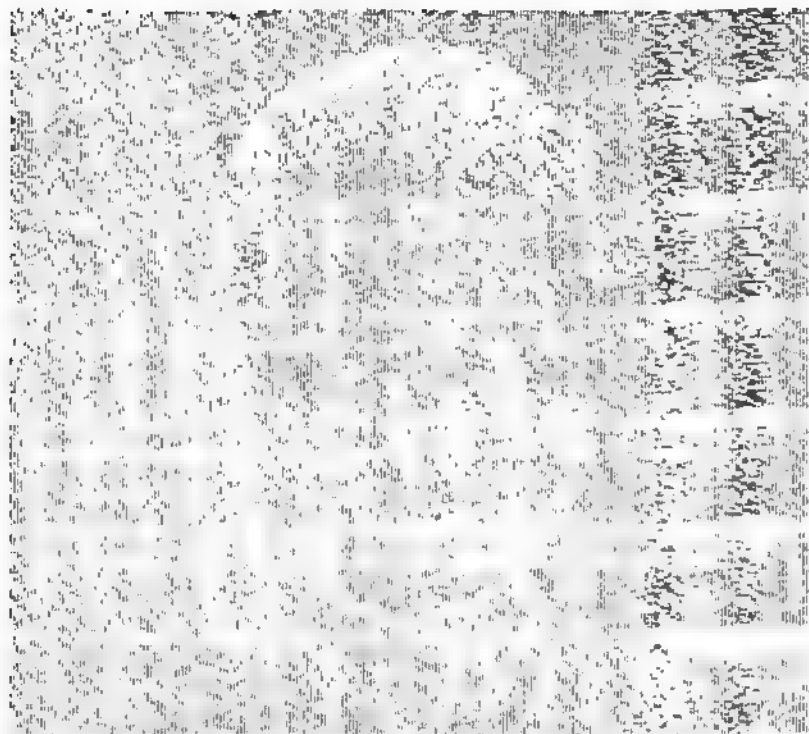



Fig. 3. *Rectangle Threshold Method.*

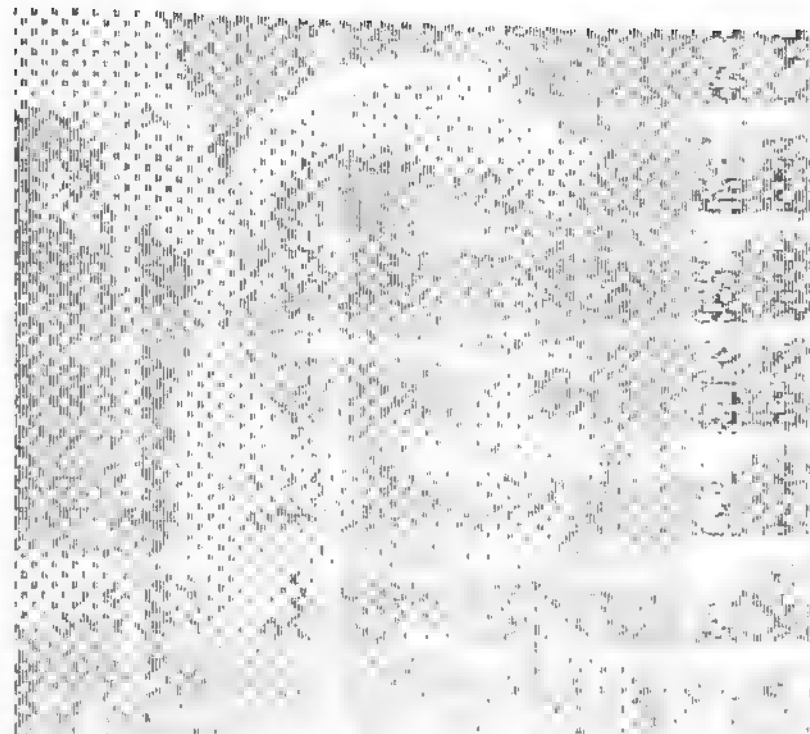



Fig. 5. *Sinusoidal Modulation Method.*

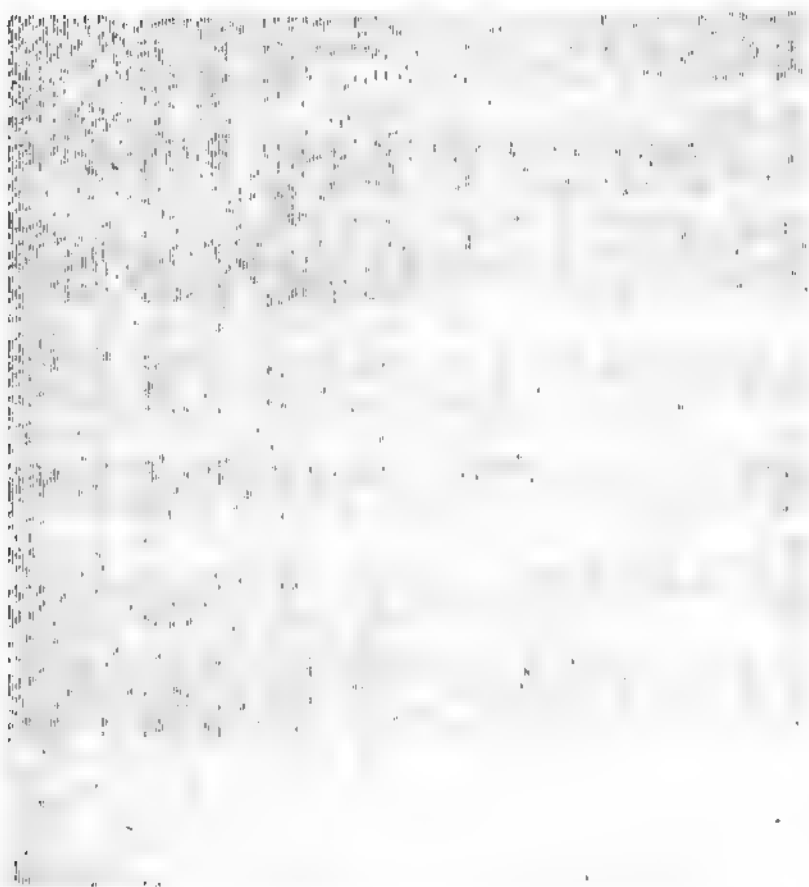



Fig. 4. *Random Modulation Method.*

Fig. 6. *Error Distribution Method.*

I tra 0 e K, K−1 dei K punti sono neri e I bianchi. Le strutture vengono scelte in modo da evitare una trama eccessiva quando vengono messe una vicine all'altra. K+1 è molto spesso 10, in quanto dieci livelli di grigio sono necessari per visualizzare la maggior parte delle immagini naturali (figura 7).

Conclusione

Le figure da 1 a 7 sono esempi di applicazioni dei metodi per le mezzetinte discussi. Le risoluzioni delle

immagini sono basse per meglio illustrare le differenze dei metodi. Molti altri metodi sono stati sviluppati per far fronte a particolari situazioni.

In bibliografia potete trovare ulteriori informazioni.

Fig. 7. *Pattern Method.*

Bibliografia

J.F. Jarvis, C.N. Judice e W.H. Ninke "A Survey of Techniques for the Display of Continuous-tone Pictures of Bilevel Displays" *Computer Graphics Image Proc.* 5, 1976.

R.J. Klensh "Electronically Generated Halftone Pictures" *RCA Review* sett. 1970.

P.G. Roetling "Halftone Method with Edge Enhancement and Moire Suppression" *J. Opt. Soc. Amer.* 66,10 ott. 1976.

R.L. Gard "Digital Picture Processing Techniques for the Publishing Industry" *Computer Graphic Image Proc.* 5, 1976.

C.N. Judice, J.F. Jarvis e W.H. Ninke "Using Ordered Dither to Display Continuous Tone Pictures" *Proc. SID* 15,4, 1974.

J.F. Jarvis e C.S. Roberts "A New Technique for the Bilevel Redition of Continuous-tone Pictures" *IEEE Trans. on Comm.* 1976.

C.N. Judice "Dithervision-A Collection of Techniques for Displayng Continuous Tone Still and Animated Pictures on Bilevel Displays" *SID*, 1975.

R.W. Floyd e L. Steinberg "An Adaptive Algorithm for Spatial Gray Scale" *Proc. SID* 17,2, 1976.

Usare il sistema operativo CP/M

Pagg. 320
Cod. 510P
L. 22.000

IL LIBRO
Il sistema operativo CP/M è stato progettato per rendere semplice l'uso di un microcomputer. Questo libro vi renderà semplice l'uso del CP/M. (Le versioni esaminate del CP/M sono il CP/M 1.4-il CP/M 2.2. e il nuovo sistema operativo multiutente MP/M) La maggior parte di utenti di microcomputer dovrà, infatti, un giorno o l'altro, fare ricorso al CP/M, disponibile su quasi tutti i computer basati sui microprocessori 8080 e Z80, come pure su certi sistemi utilizzanti il 6502. Il libro, senza presupporre alcuna conoscenza di un calcolatore, inizia con la descrizione, passo-passo delle procedure di inizializzazione del sistema: accensione, inserimento dei dischetti, esecuzione delle più comuni operazioni su file, compresa la duplicazione dei dischetti. Prosegue con il PIP (programma di trasferimento dei file), il DDT (programma di messa a punto) e ED (programma editor). Per entrare sempre più, fornendo numerosi consigli pratici, all'interno del CP/M e delle sue operazioni, al fine di comprenderne appieno le risorse ed eventualmente dare gli strumenti per successive modifiche.

SOMMARIO
Introduzione al CP/M e all'MP/M-Le caratteristiche del CP/M e dell'MP/M-Gestione dei file con PIP-L'uso dell'editor-Dentro al CP/M e all'MP/M-Guida di riferimento ai comandi e ai programmi del CP/M e dell'MP/M-Consigli pratici-Il futuro-messaggi comuni di errore-tabella di controllo di ED-nomi dei dispositivi di PIP-riassunti dei comandi-parole chiave di PIP-parametri di PIP-tasti di controllo per la digitazione dei comandi-tipi di estensione-lista dei materiali-organizzazione della stanza del calcolatore-verifiche in caso di errore-regole di base per la localizzazione dei guasti.

Per ordinare i volumi utilizzate l'apposito tagliando inserito in fondo alla rivista.

CP/M CON MP/M

GRUPPO EDITORIALE JACKSON Divisone Libri

Tutti i termini dell'informatica

Giuseppe Staluppi

Otto Vollnhals
Dizionario di informatica
Milano: Gruppo Editoriale Jackson, 1982
905 pagine, 45.000 lire

Il libro di cui parliamo in questo numero, non è di software, ma può interessare ugualmente ai "softwaristi". Si tratta di un dizionario trilingue, italiano, inglese e tedesco di termini dell'informatica. È stato compilato da un autore tedesco con l'aiuto delle tecniche informatiche più avanzate: i termini sono stati memorizzati elettronicamente, elaborati e selezionati secondo i diversi aspetti linguistici mediante un calcolatore elettronico utilizzando il programma TEAM della Siemens AG di Monaco di Baviera e composti elettronicamente con un fotocompositore DIGISET.

Il dizionario è diviso in tre parti: la prima è una parte lessicografica trilingue (inglese, italiano, tedesco) ordinata alfabeticamente secondo i termini inglesi. Questa parte comprende l'informazione terminologica completa composta dalle voci principali (lemmi, traduzioni, sinonimi, indicazioni del genere, spiegazioni), dai rimandi, dalle abbreviazioni e dagli acronimi.

La seconda e terza parte sono invece due registri alfabetici: rispettivamente dall'italiano all'inglese e dal tedesco all'inglese. Queste due parti rimandano alle corrispondenti voci della prima parte.

I settori trattati sono tutti quelli dell'informazione: hardware, software, elaborazione dell'informatica, organizzazione, micro-processori, minicomputer, midicomputer, elaborazione dei testi, tecnica circuitale e altri settori collaterali, come l'elettronica, la cibernetica, il controllo numerico. I termini riportati sono circa 15000.

Abbiamo verificato a caso qualche voce. *Bug* è tradotto con errore di programmazione, *polling* con interrogazione ciclica, *editing* con editazione e *reset* con azzeramento. Ci sembrano buone traduzioni, che rispettano il senso più che la lettera del termine originale. Ci sentiamo quindi di consigliare questo dizionario a chi ha problemi di traduzione, ma anche a tutti coloro che sono interessati al linguaggio dell'informatica. Potrà essere utile per risolvere diversi problemi lessicali, molto frequenti quando si è alle prese con un gergo in continua evoluzione.

Alcuni confronti tra le traduzioni italiane e quelle tedesche.

Software è naturalmente tradotto in italiano con software, ed anche in tedesco. Qui però abbiamo anche *immaterielle Ware* che letteralmente significa "merce non materiale" e *Programmausrüstung* (dotazione di programmi).

Interessante anche bus, che viene tradotto, oltre che con bus sia in italiano che in tedesco, anche con *Sammelleitung* in quest'ultima lingua (linea di raccolta). Record, resta record in italiano, ma diventa *Eintrag* e *Satz* in tedesco (rispettivamente registrazione e composizione, blocco).

Vediamo *linkage editor:* in italiano è correlatore, oltre a numerose altre espressioni (legatore, programma di collegamento, ecc.), mentre in tedesco è *Binder* (legatore, appunto, usato anche per il rilegatore di libri). Anche interrupt resta così in italiano, ma diventa *Unterbrechung* in tedesco.

Storage è memoria in italiano e *Speicher* in tedesco. Come si vede le lingue anglosassoni preferiscono parlare in termini di "magazzino", diversamente dall'italiano.

Concludiamo rilevando come anche questa possa essere una chiave di lettura del dizionario: una analisi comparata delle varie voci nelle varie lingue, per vedere come l'influenza angloamericana sia diversamente penetrata nei linguaggi nazionali, ed in ultima analisi per scegliere il miglior modo di usare i termini dell'informatica.

Cannonate

L'obiettivo di questo gioco per una o due persone, è quello di colpire con un colpo diretto il cannone nemico. Il gioco è scritto con grafica in alta risoluzione usando differenti dislocazioni territoriali e differenti condizioni di vento ogni volta che si gioca. Il livello di difficoltà può essere selezionato per adattarsi all'abilità dei giocatori.

Come si gioca

Il tuo Apple resterà occupato per alcuni secondi e poi ti mostrerà il campo di battaglia, le due posizioni dei cannoni e una bandierina. La bandierina è visualizzata sulla destra del video. La battaglia è combattuta (1) selezionando il numero di sacchi di polvere da sparo da caricare e l'angolo di alzo dell'arma, e (2) sparando con il cannone.

La velocità del vento è visualizzata sulla parte bassa del video sotto la bandierina. La posizione della bandierina indica la direzione del vento.

L'elaboratore selezionerà quale dei giocatori comincerà il gioco e il suo nome apparirà sulla destra per il proprio giocatore e sulla sinistra per il secondo giocatore. I turni sono dati alternativamente.

A Dopo aver notato la velocità del vento, puoi scegliere di cambiare l'alzo o di caricare il cannone con un diverso numero di sacchi di polvere da sparo.

- Per cambiare l'alzo:
  IMMISSIONE DA TASTIERA: Batti la lettera "A" alzo, seguito da un numero da due cifre (angolo), poi premi "RETURN".
  PALETTE: Ruota il controllo della paletta fino a quando ti appare la giusta angolazione sullo schermo.

- Per cambiare il numero dei sacchi di polvere da sparo (velocità del proiettile):
  IMMISSIONE DA TASTIERA: Batti la lettera "P" (per polvere), seguito dal numero dei sacchi desiderato, poi batti "RETURN".
  PALETTE: Muovi il controllo della paletta (che ti mostra l'angolo) fino ad andare oltre i 127 gradi. L'angolo che ti apparirà sarà XXX. Premi il pulsante della tua paletta fino a quando ti apparirà il giusto numero di sacchi.
  NOTA: Puoi usare la tastiera per cambiare il numero dei sacchi, anche se stai giocando con le palette.

B Quando è stato aggiustato l'angolo e l'arma è stata caricata con l'appropriato numero di sacchi di polvere da sparo, si potrà fare fuoco.

- Per fare fuoco con il cannone:
  IMMISSIONE DA TASTIERA: Premere la lettera "F" (fuoco).
  PALETTE: Premere il pulsante della paletta, o premere "F".

C La traiettoria del proiettile verrà segnata sul video e il luogo della caduta verrà indicato da un flash. Se il proiettile colpisce il centro dell'altro cannone, quest'ultimo esploderà.

D Se il proiettile colpisce il cannone avversario, il giocatore vincente riceverà un punteggio basato sulla difficoltà e sul numero di colpi che sono stati necessari per colpire. Se il proiettile manca il cannone avversario, l'altro giocatore (se c'è) avrà il proprio turno. La media dei punteggi ottenuti da ogni giocatore è mostrata sul video.

E Se desisti dal colpire il cannone nemico, puoi cominciare un'altra battaglia premendo la lettera "K".

F Dopo che hai visto il punteggio puoi cominciare un'altra partita oppure finire il gioco in questo modo:
IMMISSIONE DA TASTIERA: Premi la barra dello spazio per ricominciare il gioco oppure la lettera "F" per finire il gioco.
PALETTE: Premi il pulsante della paletta 1, oppure batti la lettera "F", come sopra.
(NOTA: Una volta che hai finito il gioco, ritorni automaticamente al MENU).

Note balistiche

Un proiettile, sparato su un campo pianeggiante e in assenza di vento, raggiungerà la massima distanza con una alzo a 45 gradi. La presenza di vento e i dislivelli del terreno, porteranno a trovare distanze massime di tiro con angolazioni superiori ai 45°.

Il raggio d'azione di un vero cannone varia da tiro a tiro anche se si userà lo stesso quantitativo di polvere da sparo, e la stessa alzo.

Questa variazione è causata da lievi differenze nell'accensione della polvere da sparo, del proiettile, e da

altre piccole variazioni. Queste differenze sono simulate dal programma. Se spari un proiettile che colpisce molto vicino all'obbiettivo, puoi sparare di nuovo senza nessun cambiamento.

È quasi sempre possibile colpire l'obiettivo usando quattro sacchi di polvere da sparo. Comunque, generalmente è meglio usare il minor numero di sacchi di polvere da sparo che permetteranno di raggiungere l'obiettivo, poiché ci saranno meno variazioni e meno effetti del vento.

Possibili modifiche

Il programma è scritto in BASIC e in linguaggio macchina. I calcoli balistici, per trovare la traiettoia più reale possibile, sono scritti con una precisione di 4 byte. Anche con 4 byte di precisione e plottando i valori, i risultati sono molto sicuri. Si è usato un ciclo che ritarda questi calcoli per avere una piacevole visione della traiettoia sul video. Se desideri cambiare la velocità del proiettile, puoi farlo in questo modo:

Velocità del proiettile: Quando il programma è caricato e pronto per essere eseguito, puoi cambiare la velocità modificando i valori delle variabili che governano la gravità e l'accelerazione. AC è la variabile accelerazione, e la trovi nella riga 8000.

Il suo valore è ora posto a 0. Più grande è il valore di AC, maggiore è la forza "gravitazionale" del campo dei battaglia. Le altre variabili determinanti sono XY e YV (V per velocità).

Sono definite nella linea 1 010. Puoi cambiare i loro rispettivi valori per modificare la velocità del proiettile.

Velocità del vento: Il gioco è scritto per cominciare con un vento fra −90 e +90. Queste velocità cambierà ad ogni colpo. La variabilità dipende dal livello di difficoltà che si è selezionato. Per rendere il gioco più facile o più difficile, puoi cambiare i limiti della velocità del vento nel seguente modo:

Quando il programma è caricato e pronto per essere eseguito, cambi i limiti della velocità del vento nella riga 80. Nella riga 80, troverai la seguente equazione:

WV= INT(RND(1)*181)−90

che controlla l'assegnazione iniziale della velocità del vento.

Per cambiarlo a 121 (per esempio), devi modificare l'equazione così:

WV= INT(RND(1)*243)−121

dove 243 è 2 volte 121 + 1.

Un cambiamento analogo deve essere fatto nella riga 503. Il valore 110 (velocità massima del vento, deve essere più elevato del valore specificato nella linea 80. Il valore 100, anche nella riga 503, deve riflettere questa stessa massima velocità.

Precisione del colpo: Il gioco è scritto in modo tale che un proiettile fa centro se è dentro le 5 unità dal centro del cannone. La precisione del colpo sarà controllata nelle righe 1 200 e 1 210. La precisione può essere posta da 1 (quasi impossibile) a 9 (facile) passi dal centro.

Per cambiare la precisione, cambia il valore 5 nelle linee 1 200 e 1 210 come desideri.

apple II

```
0 REM ******************************************
      *                                        *
      *              CANNONATE                 *
      *                                        *
      *         VERSIONE APPLE II              *
      *                                        *
      *         PERSONAL SOFTWARE              *
      *                                        *
      ******************************************
1  HIMEM: 7935: GOSUB 8000: GOTO 20
2 OX = X:OY = Y
4 X = X + XV:Y = Y - YV + AZ
6 YV = YV - AC:XV = XV + W
8  HCOLOR= 0: IF OX    0 AND OY < 160 THEN  HPLOT
    OX,OY
10  HCOLOR= 3: IF X < 0 OR X   255 THEN K =
    0: RETURN
12  IF Y    0 AND Y    160 THEN  HPLOT X,Y
14  IF Y     PEEK (GT + X) THEN K = 1: RETURN

16  GOTO 2
20  INVERSE : VTAB (1): HTAB (14): PRINT  SEL
    14): VTAB (10): HTAB (14): PRINT  SE
    13): FOR I = 1 TO 10: VTAB (I): HTAB (1
    3): PRINT " ":  HTAB (27): PRINT " ": NEXT
    : NORMAL
21  VTAB 3: HTAB 16: PRINT  CANNONATE": PRINT
     PRINT   HTAB 19: PRINT "DI": PRINT : HTAB
    1 : PRINT "T.J.SCHAFER"
60  VTAB (14): INPUT "LIVELLO DI DIFFICOLTA'
     (DA 0 A 100): ";WC: IF WC    0 OR WC
    100 THEN 60
69  VTAB (16): PRINT "QUANTI GIOCATORI? (1 O
     2) ";
70  HTAB (27): GET NP: PRINT NP;: IF NP < 1 OR
    NP > 2 THEN 70
71 NP = NP - 1: PRINT : PRINT
72  FOR I = 0 TO NP
73  VTAB (18 + 2 * I): HTAB (1): PRINT "NOME
     GIOCATORE ";I + 1;: INPUT ": ";N$(I): IF
    N$(I) = "" THEN 73
74  IF  LEN (N$(I)) > 10 THEN N$(I) =  LEFT$
     N$(I),10)
75  NEXT I
76  VTAB (22): HTAB (1): PRINT "TASTIERA O P
    ALETTE? (T O P) ";: GET A$: PRINT A$: IF
    A$ = "T" THEN KY = 1: GOTO 80
77  IF A$ = "P" THEN KY = 0: GOTO 80
78  GOTO 76
80 GA(0) = 45:GA(1) = 45:OA(0) = GA(0):OA(1)
     = GA(1):WV =  INT ( RND (1) * 181) - 9
    0:NT = NT + 1:NS
85 OWV = WV
90 K =  INT ( RND (1  *      50 -  INT ( RND
     (1) * 101)
92 B(0) = 3:B(1) = 3
95  HOME
96  VTAB 12: HTAB 11: FLASH : PRINT "UN MOME
    NTO, PREGO";: NORMAL
```

```
100 H = 159 -  INT ( RND (1) * 100)
110 J = H -  INT ( RND (1) * (H - 59))
125  FOR P = 0 TO 1
127 N = 159: IF K = P THEN N = H
130  IF P THEN 132
131  IF  NOT P THEN  FOR I = 0 TO 11: POKE S
     T + I,N: NEXT I: GOTO 134
132  FOR I = 243 TO 255: POKE GT + I,N: NEXT
     I
134 S = 0
135 M = 0: FOR I = 11 TO 127:II = I: IF P THEN
     II = 255 - I
136  IF  NOT M THEN 144
140  IF  INT ( RND (1) * (((128 - I) / 16) +
     1)) OR  NOT (N - J) THEN 159
141 M =  INT ( RND (1) * 21) + 1:S =  INT ( RND
     (1) * 6): IF  NOT S THEN 147
142  IF S = 5 THEN 159
143  GOTO 147
144  IF S = 5 THEN 159
145  IF  NOT S THEN 147
146  IF I - ( INT (I / S) * S) THEN 159
147 L = 1
148 N = N - L
159 M = M - 1: POKE GT + II,N: NEXT I
160  NEXT P
270  GOSUB 6000
280 H =  - 1
300 P =  INT ( RND (1) * 2)
500 P =  NOT P:NS = NS + 1
502  IF  NOT NP THEN P = 0
503 WV = WV +  INT ( RND (1) * [illegible] * WC + 1)
     - WC: IF  ABS (WV) [illegible] 100 THEN WV = 100
     *  SGN (WV): GOTO 503
504  HOME
505  VTAB 21: HTAB [illegible] + P * 20: FLASH : PRINT
     N$(P);: HTAB 36: NORMAL : PRINT [illegible]
     : VTAB 22: HTAB 38: PRINT  ABS [illegible]
510  VTAB [illegible]: HTAB [illegible]: PRINT "[illegible]
     [illegible] HTAB 20: PRINT "BATTI [illegible]
     [illegible]
5[illegible]  FLASH : PRINT " ";: NORMAL
54[illegible]  [illegible] 6300:L1 = P:L2 = P
5[illegible]  [illegible] 16368,0
54[illegible]  VTAB [illegible]: HTAB 14: PRINT " ";
5[illegible]  VTAB 22: HTAB 14: PRINT GA(P);: GOSUB 6
     14[illegible]
5[illegible]  [illegible] PEEK ( [illegible]16384): IF T [illegible] 127 THEN 5[illegible]
[illegible]
[illegible]  [illegible] 16[illegible] + P) [illegible] 127 THEN 100[illegible]
[illegible]  [illegible] IF P THEN
     [illegible]
[illegible]  [illegible] HTAB 15: PRINT [illegible]
     [illegible] THEN [illegible]
5[illegible]  VTAB 22: HTAB 7: PRINT " ";: HTAB 7: PRIN
     [illegible]);: HTAB 14: PRINT "XXX ";:T =  PEEK
     [illegible]: FOR V = 0 TO [illegible]: NEXT V
[illegible]  IF  PEEK [illegible] 16[illegible]87 + [illegible] [illegible] 128 THEN 56
[illegible] B(P) [illegible] B(P) + 1: PRINT "": IF B(P) [illegible] 4 THEN
     B(P) [illegible] 1
568  GOTO 5[illegible]
570  VTAB 22: HTAB 34: INVERSE : PRINT  CHR$
     [illegible] 128);: NORMAL : IF T = 198 THEN 10
     [illegible]
571  [illegible] 19[illegible] THEN 600
57[illegible]  [illegible] 20[illegible] THEN 1500
573  IF [illegible] 208 THEN  VTAB 1: PRINT "": GOTO
     504
580  POKE  [illegible] 16368,0: PRINT : PRINT
590  PRINT "POLVERE DA SPARO (DA 1 A 4): ";:
     FLASH : PRINT " ";: NORMAL
591 T =  PEEK ( [illegible] 16384): IF T [illegible] 128 THEN 59
     1
592  PRINT  CHR$ (T [illegible] 128);: IF T [illegible] 177 OR T
     [illegible] 180 THEN 580
593 B(P) = T - 176: GOTO 504
600  POKE  - 16368,0: PRINT : PRINT
610  INPUT "ALZO (DA 0 A 127): ";GA(P): IF [illegible]
     [illegible] P) [illegible] 0 OR GA(P) [illegible] 127 THEN 610
620  GOTO 504
1000  POKE  - 16368,0:I = GA(P): IF P THEN [illegible]
     = 180 - I
100[illegible] T = GX%(P):GY% =  PEEK (T + GT)
100[illegible]  [illegible] 1: HTAB 1: PRINT  SPC( 34);
1010 XV = (1.6 +  RND (1) * .1) *  COS (GA(P
     ) * PC) * B(P):YV = (1.6 +  RND (1) * 0
     .1) *  SIN (GA(P) * PC) * B(P)
1020  IF P THEN XV =  - XV
1143 W = WV / 15000
1144 X = GX%(P):Y = GY%
1146  GOSUB 2
1200  IF  ABS (GX%(0) - X) < 5 THEN H = 0: GOTO
     1250
1210  IF  ABS (GX%(1) - X) < 5 THEN H = 1: GOTO
     1250
1215  IF  NOT K THEN 500
1216  IF X < 4 OR X > 255 THEN 500
1217  CALL 768
1220  FOR I = 2 TO 0 STEP  - 2: HCOLOR= I
1230  HPLOT X,Y TO X - 4,Y - 3 TO X - 1,Y -
     2 TO X,Y - 6 TO X + 1,Y - 2 TO X + 4,Y -
     3 TO X,Y
1231  FOR J = 0 TO 200: NEXT J
1235  NEXT I
1240  GOTO 500
1250  HCOLOR= 2: FOR I = 0 TO 10: CALL 768:L
     1 = X +  RND (1) * 60 - 30:L2 = Y -  RND
     (1) * 30
1260  IF L1 < 0 THEN L1 = 0
1270  IF L1 > 255 THEN L1 = 255
1280  HPLOT X,Y TO L1,L2: NEXT I
1290  IF H >  - 1 THEN HT( NOT H) = HT( NOT
     H) + 1
1300  FOR I = 0 TO 1500: NEXT I
1500 NS = (NS + 1) / (NP + 1): FOR I = 0 TO
     NP
1502 L1 = (1000 + WC * (NP + 1) * 10) / NS: IF
     H = I OR H =  - 1 THEN L1 = 0
1508 PP(I) = (PP(I) * (NT - 1) + L1) / NT
1509  NEXT I
1510  TEXT : HOME : POKE  - 16368,0
1511  FOR I = 0 TO NP:ZZ$ = "I": IF HT(I) [illegible]
     1 THEN ZZ$ = "O":
1512  VTAB 6 + 5 * I: INVERSE : PRINT N$(I):
     NORMAL : PRINT " HAI FATTO "HT(I)" CEN
     TR"ZZ$" E " INT (PP(I))" PUNTI!":T = PP
     (I) / 50: IF T > 9 THEN T = 9
1513  RESTORE : FOR J = 0 TO T: READ A$: NEXT
     J
1514  VTAB 6 + 5 * I + 2: HTAB 2: PRINT "IL
     TUO GRADO E' ";: INVERSE : PRINT A$;: NORMAL
     : PRINT "."
1515  NEXT I
1516  VTAB 22: FOR I = 0 TO 2000: NEXT I
1520  PRINT "PREMI 'F' PER FINIRE, QUALSIASI
     ALTRO  TASTO O PULSANTE PER CONTINUAR
     E: ";: FLASH : PRINT " ";: NORMAL
1525  IF  PEEK ( - 16287) > 127 OR  PEEK (
     16286) > 127 THEN 80
1530 J =  PEEK ( - 16380): POKE  - 16380,0: IF
     J < 128 GOTO 1525
1550  POKE  - 16368,0: GOTO 1600
1560  PRINT : PRINT : NEXT I
1600  IF J [illegible] 198 AND J [illegible] 70 THEN 80
1610  TEXT : HOME : PRINT : PRINT  CHR$ (4)
     EXEC SWITCH"
1620  END
6000  HGR
6005  HCOLOR= 1
6010  FOR I = 0 TO 255
6020  HPLOT I,160 TO I, PEEK (GT + I)
6030  NEXT I
6040  FOR I = 256 TO 279: HPLOT I, PEEK (GT +
     255) TO I,160: NEXT I
6050  GOSUB 6300
6100 GX%(0) = 7 +  INT ( RND (1) * 90):GX%(1
     ) = 158 +  INT ( RND (1) * 90)
6110  FOR L = 0 TO 1
6115 I = GX%(L)
6120  HPLOT I - 4, PEEK (I + GT) - 1 TO I +
     4, PEEK (I + GT) - 1 TO I + 6, PEEK (I +
     GT) - 3 TO I - 6, PEEK (I + GT) - 3 TO
     I - 4, PEEK (I + GT) - 1
6142  NEXT L:L1 = 0:L2 = 1
6143  FOR L = L1 TO L2:J = GA(L): FOR K = 0 TO
     3 STEP 3: HCOLOR= K: IF L THEN J = 180 -
     J
6144 XX = 8 *  COS (J * PC):YY = 8 *  SIN (J
     * PC)
6145  FOR V =  - 1 TO 1:I = GX%(L):T = [illegible] I +
```

```
6150  HPLOT T, PEEK (I + GT) - 3 TO T + YX, PEE[illegible]
      (I + GT) - YY - 3
6190  NEXT V:J = GA(L)
6200  NEXT R:OA(L) = GA(L): NEXT L
6210  RETURN
6300  FOR I = 0 TO 3 STEP 3
6301  HCOLOR= I
6305 K =  PEEK (GT + 255):J = 8 *  SGN (OW[illegible]

6310  HPLOT 269,K TO 269,K - 20 TO 269 + J,[illegible]
      - 20 TO 269 + J,K - 15 TO 269,K - 15
6315 OWV = WV: NEXT I
6320  RETURN
7999  END
8000  TEXT : HOME :OX = 0:OY = 0:X = 0:Y [illegible]
      :XV = 0:YV = 0:W = 0:GX%(0) = 0:GX%[illegible] =
      0:GY% = 0:AC = .08
8001 A2 = AC / 2:WV = 0:GV = 127:GT = 793[illegible]
8002  DIM X(1),Y(1),XV(1),YV(1),GA(1),OA([illegible]
      B(1),N$(1)
8010  DIM HT(1),LN(1),PP(1):PP(1) = 0:NT =
      :HT(1) = 0:PP(0) = 0:HT(0) = 0
8020 PC = 3.14159 / 180
8030  FOR I = 0 TO 9: READ A$: NEXT I
8040  FOR I = 0 TO 29: READ J: POKE 768 + I,
      J: NEXT I
8900  DATA        BORGHESE,SOLDATO,CAPORALE,SE
      RGENTE,TENENTE,CAPITANO,MAGGIORE,TEN.CO
      LONNELLO,COLONNELLO,GENERALE
8910  DATA 169,3,133,10,165,128,[illegible]
      0,240,17,160,64,173,48,192,190,[illegible]
      2,208,25[illegible],136,208,244,76,4,3,96
8999  RETURN
```


franco muzzio editore

Il personal computer come professione

Il personal computer vi offre mille opportunità di lavoro: potete scrivere articoli o libri su computer, intraprendere un'attività di consulenza o organizzare un'esposizione locale. Per ognuna di queste attività e per molte altre il libro contiene numerose indicazioni e consigli per la miglior riuscita.

L. 7.500

Te ne intendi di computer?

Lo scopo di questo libro è aumentare il livello della vostra comprensione dei computer. Sapere cosa possono e cosa non possono fare. Sapere qual è il loro ruolo nella società, sapere quali problemi creano. Dopo questa lettura, ve ne intenderete di computer.

L. 8.500

Il piacere del computer è *la prima collana interamente dedicata alle applicazioni hobbystiche e professionali del personal computer. Questi libri descrivono l'hardware e il software, insegnano la programmazione in vari linguaggi, offrono molteplici applicazioni e informazioni pratiche. Per conoscere gli altri titoli finora apparsi (relativi al PET/CBM, all'Apple, al Basic, al Pascal, al TRS-80 e ad altri argomenti) chiedete il catalogo generale a*

franco muzzio & c. editore
via bonporti 36 - 35100 padova

cognome e nome

indirizzo

cap, località

PS2

Ippica

Questo è un tipico programma di scommesse sui cavalli.

Possono partecipare fino a nove giocatori ognuno con un capitale iniziale di $ 1000. Le corse si svolgono tra cinque cavalli ognuno quotato in maniera proporzionale alle sue possibilità di vittoria e durano in media un minuto ciascuna.

Si avverte che il programma non fà credito!

Buona fortuna!

apple II

```
1 REM ********************************************
2 REM *                                          *
3 REM *                 IPPICA                   *
4 REM *                                          *
5 REM *            VERSIONE APPLE II             *
6 REM *                                          *
7 REM *            PERSONAL SOFTWARE             *
8 REM *                                          *
9 REM ********************************************
10  TEXT : HOME
15  DIM N$(27)
20  FOR I = 0 TO 27: READ N$(I): NEXT I
30  PRINT "BENVENUTI ALLE CORSE DEI CAVALLI!
    "
50  PRINT : PRINT : INPUT "NUMERO DEI GIOCAT
     ORI? ";W
90  FOR Y = 1 TO W: PRINT : PRINT "GIOCATORE
     N.";Y;: INPUT ": ";B$(Y): NEXT
110  FOR U = 1 TO W:B(U) = 1000: NEXT
130  FOR L = 1 TO 5:S =  INT (29 *  RND (1))
     + 2:R(L) = S: NEXT
150  HOME
170  INVERSE : PRINT "Q U O T E  T O T A L I
     Z Z A T O R E": NORMAL
190  FOR Y = 1 TO 5
195 N =  INT ( RND (1) * 28):SW = 0
200  FOR J = 1 TO Y - 1: IF N(J) = N THEN SW
     = 1
205  NEXT J: IF SW = 1 THEN 195
210  PRINT : HTAB 3
220 N(Y) = N
230  PRINT Y;") ";N$(N(Y));" DATO ";R(Y);" A
     1": NEXT Y
250  FOR Y = 1 TO W: PRINT : VTAB 14: PRINT
     B$(Y);" HA ";B(Y);"$"
270  PRINT : INPUT "QUANTO VUOI PUNTARE? ";P
     (Y): IF P(Y) <  = B(Y) THEN 290
280  INVERSE : PRINT "N O N  S I  F A  C R E
     D I T O": NORMAL : GOTO 270
290  PRINT : INPUT "SUL CAVALLO NUMERO? ";Z(
     Y): IF Z(Y) < 1 OR Z(Y) > 5 THEN 290
310  FOR Q1 = 14 TO 21
330  VTAB Q1: CALL  - 868
350  NEXT : NEXT
370  HOME : GR
390  COLOR= 12: FOR I = 0 TO 39: HLIN 0,39 AT
     I: NEXT I
410 A = 1:X(1) = 2:Y(1) = 2:C(1) = 2: COLOR=
     2: GOSUB 550
430 A = 2:X(2) = 2:Y(2) = 10: COLOR= 7:C(2) =
     7: GOSUB 550
450 A = 3:X(3) = 2:Y(3) = 18: COLOR= 8:C(3) =
     8: GOSUB 550
470 A = 4:X(4) = 2:Y(4) = 26: COLOR= 13:C(4)
     = 13: GOSUB 550
490 A = 5:X(5) = 2:Y(5) = 34: COLOR= 1:C(5) =
     1: GOSUB 550
510 A =  INT (5 *  RND (1)) + 1: GOSUB 630
530  GOTO 510
550  PLOT X(A) + 1,Y(A) - 1: PLOT X(A) + 2,Y
     (A) - 1: PLOT X(A) - 2,Y(A): PLOT X(A) -
     1,Y(A): PLOT X(A),Y(A): PLOT X(A) + 1,Y
     (A)
570  PLOT X(A) - 2,Y(A) + 1: PLOT X(A) - 1,Y
     (A) + 1: PLOT X(A),Y(A) + 1: PLOT X(A) +
     1,Y(A) + 1
590  PLOT X(A) - 2,Y(A) + 2: PLOT X(A) - 2,Y
     (A) + 3: PLOT X(A) + 1,Y(A) + 2: PLOT X
     (A) + 1,Y(A) + 3
610  RETURN
630  COLOR= 12: GOSUB 550:B =  INT ( RND (1)
      * 3 / R(A) ^ .25) + 1:X(A) = X(A) + B:
      IF X(A) <  = 37 THEN  COLOR= C(A): GOSUB
     550: RETURN
650 T = A
670  GOTO 930
930  TEXT : HOME : INVERSE : PRINT "* * * *
     *  VINCITE E PERDITE  * * * * *": NORMAL

950  PRINT : PRINT "IL VICITORE E' IL NUMERO
      ";T;", ";N$(N(T))
970  FOR Y = 1 TO W: PRINT : IF T = Z(Y) THEN
     L = T: PRINT B$(Y);" HA VINTO ";P(Y) *
     R(L);"$":B(Y) = B(Y) + (P(Y) * (R(L) -
     1)): GOTO 1010
990  PRINT B$(Y);" HA PERSO ";P(Y);"$":B(Y) =
     B(Y) - P(Y)
1010  NEXT : PRINT : PRINT "UN'ALTRA SCOMME
     SA (S/N)";: INPUT "? ";F$: IF F$ = "S" THEN
     130: END
2000  DATA   "NERONE"  ,"NAPOLEONE"   ,"BAIO
     ","DIOGENE","GIULIO CESARE","MUSTANG"

2010  DATA "BALZEBU'","RUBINO","CIVETTA","RE
      ARTU'","TORO SEDUTO","SAETTA"

2020  DATA   "BARBAROSSA","MERCANTE","SMERAL
     DO","CUCULO","SOMBRERO","BARBERA","STAL
     LONE"
2030  DATA   "DIVINO","SANGIOVESE","MAFIOSO"
     ,"BO DEREK","THE LAST HURRA'","MARCANTO
     NIO","PEPE","COWBOY","MENISCO"
```

Galaxia

Galaxia è una versione per APPLE II di un gioco piuttosto diffuso nei bar e nelle sale giochi.

Si tratta di un'invasione di esseri extragalattici (i crinoidi) che il giocatore deve distruggere se vuole salvare se stesso e la galassia.

Il gioco ha tre livelli di difficoltà e va giocato con le palette: con la manopola si sposta il cannone-laser e con il pulsante si spara.

apple II

```
0 REM ***********************************************
2 REM *                                             *
3 REM *                 GALAXIA                     *
4 REM *                                             *
5 REM *          VERSIONE APPLE II                  *
6 REM *                                             *
7 REM *          PERSONAL SOFTWARE                  *
8 REM *                                             *
9 REM ***********************************************
90  LOMEM: 24577: HIMEM: 31000: GOSUB 1000
100  DIM S(20,15),A(6,3)
110  TEXT : HOME :S = 19 / 265:T = 265 / 255

120  POKE 232,99: POKE 233,124: SCALE= 1: ROT=
     0: HCOLOR= 0:S1 = 31789:S2 = 31821:STAR
     = 31776:HIT = 31917:DX = 31950:DY = 31
     951:PHASER = 31952:M1 = 32322:M2 = 3232
     5:UP = 32247:LFT = 32255:DE = 32273:DM =
     32281:MOVE = 32241
130  FOR X = 0 TO 3: FOR X1 = 0 TO 6:A(X1,X)
     =  - 1: NEXT X1,X
140  VTAB 12: INPUT "GRADO DI DIFFICOLTA' (1
     -3)? ";A$:LE =  VAL (A$): IF LE < 1 OR
     LE > 3 THEN 140
150  HGR2 :SCR = 0:AN = 0:LE = 4 - LE
160  IF LE < 1 THEN LE = 1
170  POKE DY,1: POKE DX,1: CALL PHASER: CALL
     PHASER: CALL PHASER: CALL PHASER: POKE
     DX,0: POKE DY,60: CALL PHASER
180 Z = 0:ACK = 0:N2 = 0:NL = 5:N1 = 0-NL /
     2:A = 8 * ( INT (6 / LE) + 1): FOR Y =
     2 TO 8 STEP LE: FOR X = 2 TO 16 STEP 2:
     S(X,Y) = 10: XDRAW 2 AT X * 14 + 7,Y *
     10: NEXT X,Y
190  HCOLOR= 0:Z = Z + 1:P =  PDL (0) * T: DRAW
     1 AT P1,152:B = P + 6:D =  INT (P * S):
     S(S * P1,15) = 0:S(D,15) = 255: XDRAW 1
      AT P,152:P1 = P
200  IF A = 0 OR Z < > 7 *  INT (Z / 7) THEN
     270
210  FOR X1 = 2 TO 16 STEP 2:XX = XX + S(X1,
     2): NEXT : IF XX + 10 * ACK < 60 THEN 2
     30
220  GOTO 270
230  HCOLOR= 0: FOR Y = LE + 2 TO 8 STEP LE:
      FOR X = 16 TO 2 STEP  - 2: IF S(X,Y) =
     10 THEN S(X,Y) = 0: GOTO 250
240  NEXT X,Y: GOTO 280
250  FOR X1 = 16 TO 2 STEP  - 2: IF S(X1,2) =
     0 THEN S(X1,2) = 10: DRAW 2 AT 14 * (X +
     N2) + 7,10 * (Y + N1): XDRAW 2 AT 14 *
     (X1 + N2) + 7,20 + 10 * N1:XX = XX + 10
     :X1 = 2: IF XX = > 60 THEN X = 2:Y = 8

260  NEXT : GOTO 240
270  IF ACK = 0 AND A = 0 THEN LE = LE - 1: GOTO
     160
280  IF  PEEK ( - 16287) < 128 THEN 380
290  POKE DY,50: POKE DX,1: CALL PHASER: IF
     ACK = 0 THEN 330
300 NOHIT = 255: FOR X = 1 TO 5: IF A(X,2) =
     D THEN X1 = X:Y1 = A(X,3):X = 5:NOHIT =
     0
310  NEXT : IF NOHIT THEN 330
320 C = Y1 * 10: FOR X = 0 TO 3:A(X1,X) =  -
     1: NEXT :ACK = ACK - 1: CALL HIT: HCOLOR=
     3: HPLOT B,150 TO B,C: SCALE= 2: XDRAW
     2 AT 14 * D + 7,C: HCOLOR= 0: XDRAW 2 A
     14 * D + 7,C: SCALE= 1: DRAW 2 AT 14 *
     D + 7,C: HPLOT B,150 TO B,C:SCR = SCR +
     10 ^ (4 - LE): GOTO 380
330 NOHIT = 255: IF D [illegible] 2 *  INT ((D + N2
     ) / 2) [illegible] N2 OR N2 [illegible] D THEN 360
340  FOR X1 = 8 + N1 TO 2 + N1 STEP [illegible]: IF
     S(D - N2,X1 - N1) [illegible] 0 THEN X = X1:C
     10 * X:X1 [illegible] 2 + N1:NOHIT = 0
350  NEXT
360  IF NOHIT THEN  HCOLOR= 3: HPLOT B,150 TO
     B,0: HCOLOR= 0: HPLOT B,150 TO B,0: GOTO
     380
370  CALL HIT: HCOLOR= 3: HPLOT B,150 TO B,C
     : SCALE= 2: XDRAW 2 AT 14 * D + 7,C: HCOLOR=
     0: XDRAW 2 AT 14 * D + 7,C: SCALE= 1: DRAW
     2 AT 14 * D + 7,C: HPLOT B,150 TO B,C:S
     (D - N2,X - N1) = 0:A = A - 1:SCR = SCR
      + 10 [illegible] (3 - LE)
380  IF ACK = 0 THEN 550
390  FOR X = 1 TO 5: HCOLOR= 0:A [illegible]
     1 = 14 * A1 + 7:A2 = A(X,1):B2 = 10 * [illegible]
     2:A3 = A(X,2):B3 = 14 * A3 + 7:A4 = [illegible]
     ,3):B4 = 10 * A4: IF A[illegible] THEN
     410
400  NEXT : GOTO 550
410  IF D [illegible] A1 THEN X1 = 1: GOTO 440
420  IF A1 [illegible] D THEN X1 =  - 1: GOTO 44
430 X1 = 0
440  IF A2 = 16 THEN 510
450  DRAW 2 AT B3,B4: HCOLOR= 3: [illegible]
     1,B2:A(X,0) = A1 + X1:A(X,1) [illegible]
     (X,2) = A1:A(X,3) = A2: IF  INT [illegible]
     1) * 3 + 1) [illegible] 3 THEN 400
460  POKE 31951,50: POKE 31950,1: CALL 319[illegible]
     : HPLOT 14 * A1 + 14 + X1,10 * A[illegible]
     14 * A1 + 14 + X1,160: HCOLOR= 0: [illegible]
     14 * A1 + 14 + X1,10 * A2 + 10 TO 14 *
     A1 + 14 + X1,160: IF D = A1 THEN [illegible]
     480
470  GOTO 400
480  SCALE [illegible]
     [illegible]
     [illegible]
     [illegible]
     : SCALE= 1:AN = AN + 1
490  IF AN [illegible] 6 THEN  RETURN
500  GOTO 550
510  IF A1 = D THEN  GOSUB [illegible]
```

```
    ,84:ACK = ACK - 1: FOR D1 = 0 TO 3:A(X,
    D1) =  - 1: NEXT : GOTO 400
520  FOR D1 = 16 TO 2 STEP  - 2: IF S(D1,8) =
    0 THEN S(D1,8) = 10: XDRAW 2 AT 14 * (D
    1 + N2) + 7,10 * (8 + N1):A = A + 1
530  NEXT : DRAW 2 AT 83,150: FOR DL = 0 TO
    3:A(X,DL) =  - 1: NEXT : FOR D1 = 2 TO
    8 - LE STEP LE: FOR D2 = 16 TO 2 STEP
    2: IF S(D2,D1) = 0 THEN S(D2,D1) = 10: XDRAW
    2 AT 14 * (D2 + N2) + 7,10 * (D1 + N1):
    D2 = 2:D1 = 8 - LE:A = A + 1
540  NEXT D2,D1:ACK = ACK - 1: GOTO 400
550  IF AN < 6 THEN 600
560  HOME : TEXT : VTAB 3: PRINT  TAB( 14)"L
    IVELLO ";4 - LE
570  VTAB 14: PRINT  TAB( 12)"PUNTEGGIO :";S
    CR * 10
580  TEXT : VTAB 20: HTAB 9: PRINT "VUOI GIO
    CARE ANCORA?";: GET A$: PRINT : IF A$ =
    "N" THEN  END
590  RUN 100
600 ZZ = 5 * (Z -  INT (Z / 3) * 3): POKE S1
    ,ZZ: POKE S2,ZY: CALL STAR:ZY = ZZ
610  IF A = 0 THEN 190
620  IF ACK <  0 AND Z <  10 *  INT (Z /
    10) THEN 680
630 YY =  SGN ( RND (1) - .5): IF NU - YY
    0 OR NU - YY  3 OR YY = 0 THEN 630
640 XX = 2 *  SGN ( RND (1) - .5): IF NL + X
    X < 1 OR NL + XX > 10 THEN 640
650  IF  SGN (YY) =  - 1 THEN 670
660  POKE M1,238: POKE M2,238: POKE UP,10 *
    NU: POKE LFT,NL - 1: POKE DE,10 * (NU -
    YY): POKE DM,NL + XX: CALL MOVE:NU = NU
     - YY:NL = NL + XX:N1 = N1 - YY:N2 = N2
     +  SGN (XX): GOTO 680
670  POKE M1,206: POKE M2,206: POKE UP,10 *
    (NU + 8) - 1: POKE LFT,NL - 1: POKE DE,
    10 * (NU - YY + 8) - 1: POKE DM,NL + XX
    : CALL MOVE:NU = NU - YY:NL = NL + XX:N
    1 = N1 - YY:N2 = N2 +  SGN (XX)
680  POKE  - 16336, PEEK ( - 16336): POKE  -
    16336, PEEK ( - 16336): POKE  - 16336, PEEK
    ( - 16336): IF A [illegible] 3 * ( INT (6 / LE) +
    1) AND N1 [illegible] - 2 THEN 190
690  IF ACK [illegible] 4 OR LE * 5 [illegible] 2 OR N1 >  - 2 THEN
    190
700  FOR X = 1 TO 5: IF A(X,1) =  - 1 THEN A
    P = X:X = 5
710  NEXT : HCOLOR= 0: FOR X = 2 TO 16 STEP
    2: IF S(X,2) [illegible] 0 THEN  DRAW 2 AT 14 *
    (X + N2) + 7,0: XDRAW 2 AT 14 * (X + N2
    ) + 7,90:ACK = ACK + 1:A(AP,0) = X + N2
    :A(AP,1) = 11 + N1:A(AP,2) = X + N2:A(A
    P,3) = 11 + N1:S(X,2) = 0:A = A - 1:X =
    16
720  NEXT : GOTO 190
1000  GOSUB 1110
1020  HOME
1022  PRINT "I CRINOIDI, UNA RAZZA CON TECNO
    LOGIE    PIU' EVOLUTE, STA TENTANDO DI
    INVADERE  LA GALASSIA."
1030  PRINT : PRINT "IL LORO SCOPO E' DI DIS
    TRUGGERE LA VITA ANIMALE E IMPORSI COME
     RAZZA SUPERIORE "
1040  PRINT : PRINT "LA TUA MISSIONE [illegible]
    ISTRUGGERE PIU'  CRINOIDI CHE [illegible]
1050  PRINT : PRINT "ATTENZIONE!"
1055  PRINT "ALCUNI CRINOIDI SCENDERANNO [illegible]
    I"
1057  PRINT "ATTACCHERANNO."
1060  PRINT "SE ARRIVERANNO ALLA BASE [illegible]
    "
1063  PRINT "RICOMPARIRANNO CON DEI RINFOR[illegible]
    ."
1070  PRINT : PRINT "LA TUA NAVE PUO' E[illegible]
     COLPITA SOLO"
1072  PRINT "CINQUE VOLTE.": PRINT "ALLA SE
    TA SARAI DISTRUTTO E LA "
1074  PRINT "GALASSIA SARA' INVASA."
1080  VTAB 23: PRINT " [illegible] PREMI  RETURN
    ER CONTINUARE [illegible]";
1090  IF  PEEK ( - 16384) < 128 THEN 1090
1100  POKE  - 16368,0: HOME : RETURN
1110  FOR X = 0 TO 615: READ A: POKE X + 71
    44,A:B = B + A: NEXT : IF B [illegible] 50709 OR
    X <  > 616 THEN  PRINT "*** DATI ERRATI
    ***": END
1120  RETURN
1130  DATA 147,211,0,98,33,16,68,38,76,4,10,
    35,51,64,68,65,64,70,64,73,70,64,66,68,
    66,67,68,67,66,67
1140  DATA 67,65,160,0,173,45,124,105,15,141
    ,51,124,169,16,190,0,124,142,57,124,190
    ,25,124,142,58,124,141,65,67,200
1150  DATA 192,6,208,236,160,0,173,77,124,10
    5,14,141,83,124,169,0,190,0,124,142,89,
    124,190,0,124,142,90,124,141,16
1160  DATA 70,200,192,6,208,236,24,96,0,2,0,
    6,0,28,0,73,73,62,54,55,62,14,45,54,63,
    55,77,73,60,39
1170  DATA 40,5,56,36,39,60,0,73,46,45,45,77
    ,181,27,63,55,45,53,63,55,45,45,45,46,4
    6,222,27,56,60,63
1180  DATA 62,4,56,67,62,62,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,169,0,168,56,46,171,1
    24
1190  DATA 46,172,124,144,3,173,48,192,106,1
    06,106,77,172,124,10,10,10,136,152,208,
    1,96,76,177,124,0,1,1,172,206
1200  DATA 124,173,48,192,174,207,124,202,20
    8,253,136,208,244,206,207,124,208, [illegible],9
    6,0,0,0,0,0,0,0,0,0,0,0
1210  DATA 0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,[illegible],6
    4,68,72,76,80,84,88,92,64,68,72,76,80,8
    4
1220  DATA 88,92,65,69,73,77,81,85,89,[illegible]
    69,73,77,81,85,89,93,66,70,74,78,[illegible],8
    90,94,66,70,74,78
1230  DATA 82,86,90,94,[illegible],71,75,79,83,[illegible]
    95,[illegible],71,75,79,83,[illegible],95,64,68,[illegible]
    80,84,88,92,64,68
1240  DATA 72,76,80,84,88,92,65,69,7[illegible]
    85,89,93,65,69,[illegible],77,81,85,89,93,[illegible]
    74,78,82,86,90,94
1250  DATA 66,71,74,78,[illegible],86,90,94,67,71,[illegible]
    79,8[illegible],0,0,0,1,8,1
    128,128,128,128
1260  DATA 128,1[illegible],0,0,0,0,0,1,8,1[illegible]
    [illegible],128,128,128,1[illegible],0,0,0,0,[illegible]
    128,128,128,[illegible]
1270  DATA 128,128,128,1,8,[illegible],1
    28,128,128,128,1,8,128,128,12[illegible],1[illegible]
    ,40,40,40,40,40,168,1[illegible]
1280  DATA 168,168,168,168,168,168,40,[illegible]
    40,40,40,40,40,168,168,168,168,[illegible]
    168,168,40,40,40,40,40,40,40,[illegible]
1290  DATA 168,168,168,168,168,1[illegible]
    [illegible],40,40,40,40,40,40,40,0,162,0,[illegible]
    [illegible],150,10,190,0,125,185,120,1,[illegible]
1300  DATA 105,2,144,3,232,24,141,41,1,[illegible]
    ,48,126,142,42,126,142,49,126,140,[illegible]
    [illegible],0,125,185,120,125,105,1,144,.
1310  DATA 272,24,141,44,126,142,45,1[illegible]6,[illegible]
    0,160,0,185,171,72,153,1,81,178,1[illegible]
    ,72,200,192,30,208,241,24,278,240
1720  DATA  125,169,80,205,240,125,240,9,[illegible]
    ,247,125,206,17,126,76,216,125,24,96
    0,0,126,76,244,1,5[illegible]
1730  DATA [illegible],48,48,48,48,54,5[illegible],47,55,[illegible]
    ,5[illegible],90,[illegible]
```

Riflessioni

Per giocare dovete conoscere bene le leggi di riflessione e di diffrazione (non temete: il programma vi spiega tutto ciò che dovete sapere). Il computer, poi, nasconde 3, 4 o 5 palline in una scatola nera.

Dovete determinare la loro posizione senza poter guardare nella scatola. Per farlo mandate dei raggi di luce attraverso le pareti della scatola: all'interno essi vengono riflessi o diffratti dalle palline nascoste. Confrontando i punti dove entrano ed escono i raggi, potrete dedurre la posizione delle palline.

Il computer calcola il punteggio finale in questa maniera: ogni raggio che urta una pallina o viene riflesso vale un punto, ognuno che entra ed esce senza incontrare ostacoli ne vale due e, infine, ogni posizione non indovinata ne fa perdere cinque. Vince il punteggio minore.

Tandy Radio Shack

```
1 REM ****************************************
2 REM *                                      *
3 REM *             RIFLESSIONI              *
4 REM *       VERSIONE TRS-80 MOD. I         *
5 REM *                                      *
6 REM *          PERSONAL SOFTWARE           *
7 REM *                                      *
8 REM ****************************************
100 CLS: PRINT@22,"RIFLESSIONI": PRINT
150 CLEAR 200
160 L$=STRING$(32,".")
180 INPUT"VUOI LE ISTRUZIONI";A$
185 CLS
190 IF LEFT$(A$,1)="N" THEN 1200: PRINT
195 PRINT: PRINT
220 PRINT"QUESTO GIOCO ESERCITA IL TUO "
222 PRINT"RAGIONAMENTO DEDUTTIVO."
224 PRINT"IL COMPUTER NASCONDE 3, 4 O 5 ";
226 PRINT"PALLINE (A TUA SCELTA) IN UNA "
228 PRINT"SCATOLA NERA."
230 PRINT"DEVI SCOPRIRE LA LORO COLLOCAZIONE";
232 PRINT" SU UNA GRIGLIA DI RIFERIMENTO";
234 PRINT"SENZA POTER GUARDARE NELLA SCATOLA."
238 PRINT"LO PUOI FARE MANDANDO RAGGI ";
240 PRINT"ATTRAVERSO LE PARETI DELLA SCATOLA."
242 PRINT"OSSERVANDO COME QUESTI VENGONO ";
244 PRINT"RIFLESSI E DIFFRATTI COLPENDO LE"
246 PRINT"PALLINE E CONFRONTANDO I PUNTI";
248 PRINT" DOVE ENTRANO ED ESCONO, POTRAI"
250 PRINT"SCOPRIRE DOVE SONO NASCOSTE."
252 PRINT"ORA VIENE DATA UNA DIMOSTRAZIONE ";
254 PRINT"DELLE LEGGI DI RIFLESSIONE E"
256 PRINT"DIFFRAZIONE."
257 PRINT
258 INPUT"SE SEI PRONTO PREMI =ENTER=";A$
260 CLS: GOSUB 2420
262 PRINT@832,"ECCO LA SCATOLA. I NUMERI ";
264 PRINT"INDICANO I PUNTI DI INGRESSO E ";
266 PRINT"USCITA DEI RAGGI. ";
268 GOSUB 10000: INPUT A$
280 PRINT@832,CHR$(31);
290 PRINT@832,"UN RAGGIO CHE ENTRA IN 4 ";
292 PRINT"E NON INCONTRA OSTACOLI, ESCE ";
294 PRINT"IN 21. "
296 GOSUB 10000
300 FOR X=20 TO 84: Y=18: SET(X,Y): FOR I=1
    TO 15:NEXT I,X
310 INPUT A$: PRINT@832,CHR$(31)
320 PRINT@832,"NATURALMENTE NEL GIOCO NON ";
322 PRINT"VEDRAI LA TRAIETTORIA."
326 PRINT"LE LETTERE INDICANO LA SEQUENZA";
328 PRINT" DEI RAGGI: ";
330 PRINT"A PER IL PRIMO,"
332 PRINT"B PER IL SECONDO, ECCETERA. ";
334 GOSUB 10000
336 FOR I=1 TO 25: POKE 15748,32: POKE 15792
    ,32: FOR K=1 TO 25: NEXT K
340 POKE 15748,65: POKE 15792,65: FOR J=1 TO
    25: NEXT J,I
350 INPUT A$
360 CLS: GOSUB 2420
370 POKE 15792,48
380 PRINT@832,"QUANDO UN RAGGIO INCONTRA ";
382 PRINT"UNA PALLINA DIRETTAMENTE, LO SI"
384 PRINT"INDICHERA' CON UNA U (URTO) NEL ";
386 PRINT"PUNTO DI INGRESSO."
388 GOSUB 10000
390 FOR X=20 TO 52: Y=18: SET(X,Y): FOR I=1
    TO 15: NEXT I,X
400 POKE 15748,85
410 INPUT A$
420 CLS: GOSUB 2420
430 POKE 15876,48
440 PRINT@832,"GENERALMENTE, SE UN RAGGIO ";
442 PRINT"VEDE UNA PALLINA SU UNO DEI DUE"
444 PRINT"LATI DEL SUO TRAGITTO, VIENE ";
445 PRINT"DEVIATO ED ESCE."
446 GOSUB 10000
450 FOR X=20 TO 49: Y=18: SET(X,Y): FOR I=1
    TO 5: NEXT I,X
460 FOR Y=17 TO 7 STEP -1: X=49: SET(X,Y):
    FOR I=1 TO 25: NEXT I,Y
470 POKE 15748,65: POKE 15384,65
480 INPUT A$
490 PRINT@758, MF$(1): PRINT@768,"N H
49  PRINT"LA DEVIAZIONE AVVIENE PRIMA CHE ".
4   PRINT" IL RAGGIO INCONTRI LA RIGA"
4   PRINT"DELLA PALLINA. ";: GOSUB 10000
5   INPUT A$
510 CLS: GOSUB 2420
520 PRINT@832,"ECCEZIONE: QUANDO VEDE UNA ".
5   PRINT" PALLINA ESATTAMENTE DI FRONTE,
524 PRINT"SI HA UN URTO. ";
526 GOSUB 10000
530 POKE 15772,48: POKE 15836,48
540 FOR X=20 TO 52: Y=18: SET(X,Y): FOR I=1
    TO 15: NEXT I,X
550 POKE 15748,85
560 INPUT A$
570 CLS: GOSUB 2420
580 PRINT@832,"UN RAGGIO CHE HA DI
582 PRINT"FRONTE UNA PALLINA DA CIASCUN ";
```

```
584 PRINT"LATO DELLA"
585 PRINT"TRAIETTORIA VIENE ";
586 PRINT"RESPINTO ALL'INGRESSO ED ";
587 PRINT"INDICATO CON R"
588 PRINT"(RIFLESSO). ";
589 GOSUB 10000
590 POKE 15836,48: POKE 15708,48
600 FOR X=20 TO 52: Y=18: SET(X,Y): FOR I=1
    TO 15: NEXT I,X
610 FOR X=20 TO 51: Y=18: RESET(X,Y)
612 NEXT X
620 FOR X=51 TO 20 STEP -1: SET(X,Y): FOR
    I=1 TO 15: NEXT I,X
630 POKE 15748,82
640 INPUT A$
650 CLS: GOSUB 2420
659 PRINT@768,CHR$(31)
660 PRINT@768,"UNA RIFLESSIONE AVVIENE ANCHE";
662 PRINT" SE CI SONO PALLINE DA UNO"
664 PRINT"DEI LATI ALL'INGRESSO: ";
666 PRINT"IL RAGGIO NON PUO' ENTRARE."
668 GOSUB 10000
670 POKE 15820,43
680 FOR X=20 TO [illegible]: Y=18: SET(X,Y): NEXT
690 POKE 15748,82
700 INPUT A$
705 PRINT@768,CHR$(31)
706 PRINT@832,"SE C'E' UNA PALLINA ";
707 PRINT"PROPRIO DI FRONTE, E' UN URTO "
708 GOSUB 10000: POKE 15756,48
709 POKE 15748,85: INPUT A$
710 CLS: GOSUB 2420
720 PRINT@832,"UN RAGGIO PUO' ESSERE ";
722 PRINT"DEVIATO PIU' DI UNA VOLTA."
724 GOSUB 10000
730 POKE 158[illegible],45: POKE 155[illegible],48
740 FOR X=20 TO 48: Y=18: SET(X,Y): FOR I=1
    TO 15: NEXT I,X
750 FOR Y=18 TO 1[illegible] STEP -1: X=48: SET(X,Y):
    FOR I=1 TO 15: NEXT I,Y
760 FOR X=48 TO 84: Y=1[illegible]: SET(X,Y): FOR
    TO 15: NEXT I,X
770 POKE 15748,65: POKE 15664,65
780 INPUT A$
790 PRINT@832,CHR$(31)
800 PRINT@832,"NOTA CHE IL RAGGIO POTREBBE";
802 PRINT" ESSERE ENTRATO DA 4 O DA 23."
804 GOSUB 10000
810 INPUT A$
820 CLS: PRINT@128,"GUARDA I PROSSIMI ";
821 PRINT"ESEMPI."
822 PRINT"SE NON TI E' CHIARO IL PERCHE' ";
824 PRINT"DELLE TRAIETTORIE, DEGLI URTI O"
825 PRINT"DELLE RIFLESSIONI, ";
826 PRINT"RILEGGI LE ISTRUZIONI."
828 GOSUB 10000
830 INPUT A$
840 CLS: GOSUB 2420
850 POKE 15836,48: POKE 15572,48: POKE
    15592,48
860 FOR X=20 TO 48: Y=18: SET(X,Y):FOR I=
    TO 15: NEXT I,X
870 FOR Y=18 TO 12 STEP -1: SET(X,Y): FOR
    I=1 TO 15: NEXT I,Y
880 FOR X=48 TO 72: Y=12: SET(X,Y):FOR I=1
    TO 15: NEXT I,X
890 FOR Y=12 TO 31: X=72: SET(X,Y): FOR I=1
    TO 15: NEXT I,Y
900 POKE 15748,65: POKE 151[illegible],65
910 PRINT@832,CHR$(31): GOSUB 10000
920 INPUT A$
930 CLS: GOSUB 2420
940 POKE 15836,48: POKE 15640,48
950 FOR X=20 TO 48: Y=18: SET(X,Y): FOR I=1
    TO 15: NEXT I,X
960 FOR Y=18 TO 15 STEP -1: X=48: SET(X,Y):
    FOR I=1 TO 15: NEXT I,Y
970 POKE 15748,85
980 PRINT@832,CHR$(31): GOSUB 10000
990 INPUT A$
1000 CLS: GOSUB 2420
1010 POKE 15836,48: POKE 15572,48: POKE
     15592,48: POKE 15720,48
1020 FOR X=20 TO 48: Y=18: SET(X,Y): FOR I=1
     TO 15: NEXT I,X
1030 FOR Y=18 TO 12 STEP -1: X=48: SET(X,Y):
     FOR I=1 TO 15: NEXT I,Y
1040 FOR X=48 TO 72: Y=12: SET(X,Y): FOR I=1
     TO 15: NEXT I,X
1050 FOR X=20 TO 48: Y=18: RESET(X,Y): NEXT
1060 FOR Y=18 TO 12 STEP -1: X=48: RESET
     (X,Y): NEXT
1070 FOR X=48 TO 71: Y=12: RESET(X,Y): NEXT
1080 FOR X=71 TO 48 STEP -1: Y=12: SET(X,Y):
     FOR I=1 TO 15: NEXT I,X
1090 FOR Y=12 TO 18: X=48: SET(X,Y): FOR I=1
     TO 15: NEXT I,Y
1100 FOR X=48 TO 20 STEP -1: Y=18: SET(X,Y):
     FOR I=1 TO 15: NEXT I,X
1110 POKE 15748,82
1120 PRINT@832,"VUOI LE ISTRUZIONI";
1130 INPUT A$
1140 IF LEFT$(A$,1)="S" THEN 220
1145 CLS
1150 PRINT"GIOCANDO PUOI METTERE LE PALLINE";
1152 PRINT" DOVE PENSI SI TROVINO."
1154 PRINT"PUOI ANCHE TOGLIERLE SE CAMBI IDEA."
1160 PRINT"SE PENSI DI AVER INDOVINATO TUTTE ";
1170 PRINT"LE POSIZIONI CORRETTE SCRIVI"
1171 PRINT"FINE."
1172 PRINT"APPARIRANNO LE PALLINE NELLE ";
1174 PRINT"POSIZIONI ESATTE E IL PUNTEGGIO."
1176 PRINT
1177 PRINT"OGNI URTO O RIFLESSIONE = 1 PUNTO"
1178 PRINT"OGNI RAGGIO CHE ESCE     = 2 PUNTI":
     PRINT"OGNI POSIZIONE ERRATA   = 5 PUNTI":
     PRINT
1179 PRINT"VINCE IL PUNTEGGIO MINORE !!!":
     PRINT: GOSUB 10000
1180 INPUT A$
1200 CLS: RANDOM: EY=64
1240 INPUT"QUANTE PALLINE (DA 3 A 5)";BN
1250 FOR I=1 TO BN
1260 X=RND(8): Y=RND(8)
1270 IF A(X,Y)=0 THEN A(X,Y)=1: GOTO 1290
1280 GOTO 1260
1290 NEXT
1300 CLS
1310 GOSUB 2420
1320 GOSUB 2340
1330 PRINT@842,"SCRIVI L'ENTRATA DEL RAGGIO";:
     INPUT E
1340 IF E>32 THEN PRINT@832,"ERRORE":
     FOR I=1 TO 500: NEXT: GOTO 2020
1350 CT=CT+1
1360 GOTO 1380
1370 REM
1380 IF E>0 AND E<9 THEN X=0: Y=E: D$="EST":
     GOTO 1440
1390 IF E>8 AND E<17 THEN X=E-8: Y=9: D$=
     "NORD": GOTO 1440
1400 IF E>16 AND E<25 THEN X=9: Y=25-E:
     D$="OVEST": GOTO 1440
1410 IF E>24 AND E<33 THEN X=33-E: Y=0:
     D$="SUD": GOTO 1440
1420 GOTO 1330
1430 REM
1440 [illegible]
1450 IF D$="EST" THEN I=1: J=0: GOTO 1490
1460 IF D$="OVEST" THEN I=-1: J=0: GOTO 1490
1470 IF D$="SUD" THEN I=0: J=1: GOTO 1490
1480 IF D$="NORD" THEN I=0: J=-1: GOTO 1490
1490 [illegible]A(X+I-J,Y+I+J)
1500 C=A(X+I,Y+J)
1510 R=A(X+I-J,Y+I+J)
1520 REM
1530 IF L=0 AND C=0 AND R=0 THEN X=X+I: Y=Y+J
     : GOTO 1730
1540 REM
1550 IF C=1 THEN 1860
1560 REM
1570 IF X=0 OR X=9 OR Y=0 OR Y=9 THEN [illegible]
1580 REM
1590 REM
1600 IF D$="EST" THEN 1640
1610 IF D$="OVEST" THEN 1660
1620 IF D$="SUD" THEN 1680
1630 GOTO 1700
1640 IF R=1 THEN D$="NORD" ELSE D$="SUD":
     GOTO 1450
```

```
1650 GOTO 1450
1660 IF R=1 THEN D$="SUD" ELSE D$="NORD":
     GOTO 1450
1670 GOTO 1450
1680 IF R=1 THEN D$="EST" ELSE D$="OVEST":
     GOTO 1450
1690 GOTO 1450
1700 IF R=1 THEN D$="OVEST" ELSE D$="EST":
     GOTO 1450
1710 GOTO 1450
1720 REM
1730 IF X=0 OR X=9 OR Y=0 OR Y=9 THEN 1750
1740 GOTO 1450
1750 EX=10*X+Y
1760 IF EX=EP THEN 1880
1770 IF X=0 THEN EX=Y
1780 IF X=9 THEN EX=25-Y
1790 IF Y=0 THEN EX=33-X
1800 IF Y=9 THEN EX=X+8
1810 CT=CT+1
1820 EY=EY+1: X=EX: I=EY: GOSUB 1900
1830 IF EY=71 THEN EY=72
1840 X=E: GOSUB 1900
1850 GOTO 2020
1860 X=E: I=85: GOSUB 1900
1870 GOTO 2020
1880 X=E: I=82: GOSUB 1900
1890 GOTO 2020
1900 IF X>0 AND X<9 THEN 1940
1910 IF X>8 AND X<17 THEN 1960
1920 IF X>16 AND X<25 THEN 1980
1930 IF X>24 AND X<33 THEN 2000
1940 POKE 15492+64*X,I: RETURN
1960 POKE 16139+4*(X-9),I: RETURN
1980 POKE 16048-(X-17)*64,I: RETURN
2000 POKE 15371+(32-X)*4,I: RETURN
2020 PRINT@832,"BATTI T PER TOGLIERE OPPURE ";
2022 PRINT"A PER AGGIUNGERE PALLINE OPPURE "
2024 PRINT"INSERISCI L'ENTRATA DI UN RAGGIO";:
     INPUT G$
2030 PRINT@832,STRING$(64," ")
2040 T$=LEFT$(G$,1)
2050 FL=0
2060 IF G$="T" THEN 2100
2070 IF T$="A" THEN 2110
2080 IF LEFT$(T$,1)="F" THEN 2270
2085 IF ASC(LEFT$(T$,1))>65 THEN PRINT@832,
     "ERRORE": FOR I=1 TO 500: NEXT: GOTO 2020
2090 E=VAL(G$): GOTO 1340
2100 FL=1
2110 PRINT@832,"COORDINATE DELLA PALLINA ";
2112 PRINT"(ES. COL26, RIGA5 = 26,5)"
2114 INPUT N1,N2
2120 PRINT@832,STRING$(64," ")
2130 NB=N1
2140 IF NB>0 AND NB<9 THEN Y=NB
2150 IF NB>8 AND NB<17 THEN X=NB-8
2160 IF NB>16 AND NB<25 THEN Y=25-NB
2170 IF NB>24 AND NB<33 THEN X=33-NB
2180 IF NB=N2 THEN 2200
2190 NB=N2: GOTO 2140
2200 IF FL=1 THEN 2230
2210 IF B(X,Y)=0 THEN B(X,Y)=1: M=43: GOTO 2250
2220 GOTO 2110
2230 IF B(X,Y)=1 THEN B(X,Y)=0: M=46: GOTO 2250
2240 GOTO 2110
2250 POKE 15496+64*Y+4*X,M
2260 GOTO 2020
2270 FOR X=1 TO 8
2280 FOR Y=1 TO 8
2290 IF A(X,Y)=1 THEN POKE 15496+64*Y+4*X,[illegible]
2300 IF A(X,Y)=1 AND B(X,Y)<>1 THEN CT=CT+5
2310 NEXT Y,X
2320 PRINT@842,"IL TUO PUNTEGGIO E' ";CT
2330 END
2340 POKE 15541,80: POKE [illegible]
     ,82: POKE 15544,[illegible]: POKE [illegible]
     [illegible]
2342 POKE 15606,[illegible]: POKE [illegible]
     [illegible]
2350 POKE 15667,8[illegible]: POKE [illegible]
     [illegible]
     [illegible]
2360 POKE [illegible]
     [illegible]: POKE 15[illegible],6[illegible]
2370 IF BN=[illegible] THEN POKE 15861,[illegible]
2380 IF BN=4 THEN POKE 15861,[illegible]
2390 IF BN=[illegible] THEN POKE 15861,[illegible]
2400 POKE 15925,80: POKE [illegible]
     ,76: POKE 15928,7[illegible]: POKE [illegible]
     15930,78: POKE 15931,[illegible]
2410 RETURN
2420 FOR X=0 TO 7
2430 PRINT@202+64*X,[illegible]
2440 NEXT
2450 FOR X=49 TO 5[illegible]
2460 POKE 15494+64*(X-48),X
2470 NEXT
2480 [illegible]
2490 [illegible]
2500 FOR X=[illegible] TO 84 [illegible]
2510 FOR Y 8 TO [illegible]
2520 SET(X,Y)
2530 NEXT Y,X
2540 POKE 160[illegible],5
2550 [illegible]
2560 FOR X=1607[illegible] TO [illegible]
2570 Y=Y+1
2580 POKE X,4[illegible]
2590 POKE X+1,Y+4
2600 NEXT
2610 FOR Y=0 TO [illegible]
2620 POKE [illegible]
2630 POKE [illegible]
2640 POKE [illegible]
2650 POKE [illegible]
2660 NEXT
2670 FOR [illegible]
2680 POKE [illegible]
2690 POKE [illegible]
2700 POKE [illegible]
2710 POKE [illegible]
2720 NEXT
2730 RETURN
10000 [illegible]
10010 RETURN
```

Bombardiere

Il gioco comincia con la vostra scelta del canyon tra quattro diversi tipi. Potete poi volare sopra di esso, sganciando bombe da un aereo (con il tasto spazio) per eliminare gli ostacoli.

Potete sganciare una sola bomba per volta e, prima di lanciarne un'altra, dovrete aspettare che la prima abbia finito il suo effetto distruttivo. Se non sganciate almeno una bomba per volo, perdete un punto.

Il gioco termina dopo tre errori.

Tandy
Radio Shack

```
1 REM ***************************************************
2 REM *                                                 *
3 REM *                  BOMBARDIERE                    *
4 REM *            VERSIONE TRS-80 MOD. I               *
5 REM *                                                 *
6 REM *               PERSONAL SOFTWARE                 *
7 REM *                                                 *
8 REM ***************************************************
10 CLS: PRINT@22,"BOMBARDIERE"
18 PRINT
20 PRINT"IN QUESTO PROGRAMMA STAI RIPULENDO "
25 PRINT"UN CANYON DAGLI OSTACOLI"
30 PRINT"SGANCIANDO BOMBE DA UN AEROPLANO."
32 PRINT"IL GIOCO CONTINUA FINCHE' NON SBAGLI";
35 PRINT" TRE VOLTE."
40 PRINT"PUOI SGANCIARE SOLO UNA BOMBA ALLA ";
45 PRINT"VOLTA, E DEVI ASPETTARE"
47 PRINT"CHE ABBIA FINITO IL SUO COMPITO PRIMA";
50 PRINT" DI SGANCIARE LA PROSSIMA."
60 PRINT"SE NON SGANCI ALMENO UNA BOMBA ";
62 PRINT"PER VOLO, PERDI UN TURNO."
63 PRINT
65 INPUT"PREMI ENTER PER GIOCARE";A$
67 CLS: CLEAR 500: A$=CHR$(141)+CHR$(140)+
   CHR$(174)+CHR$(140)
70 B$=CHR$(140)+CHR$(157)+CHR$(140)+CHR$(142)
80 INPUT"CHE CANYON VUOI (1-4)";A: CLS:
   IF A=3 THEN 550
85 IF A=4 THEN 600
90 IF A=2 THEN 500
100 IF A<>1 THEN 80
110 FOR X=0 TO 8: PRINT@X*64+384,STRING$
    (3+X*3,191);: NEXT
112 PRINT@896,STRING$(63,191);: PRINT@960,
    STRING$(63,191);
115 FOR X=0 TO 1: PRINT@817+X*64+(2-X)*3,
    STRING$(8+X*3,191);: NEXT
120 FOR X=442 TO 762 STEP 64
122 PRINT@X,STRING$(5,191);: NEXT
130 FOR X=-1 TO 8: FOR Y=1 TO 19
133 IF PEEK(15681+X*64+Y*3)=191 THEN POKE
    15681+X*64+Y*3,188: POKE 15681+X*64+Y*3+1
    ,188
140 NEXT Y,X: PRINT@0,"CENTRI";0;: PRINT@32,
    "ERRORI";0;
150 H=1
160 S=RND(0)+.5: IF D=1 THEN S=-S
170 FOR X=0+D*60 TO 60-D*60 STEP S
172 PRINT@H*64,CHR$(30);
175 IF D=0 THEN PRINT@H*64+X,A$;: GOTO 190
180 PRINT@H*64+X,B$;
190
    IF INKEY$=" " AND O=0 THEN O=1: X1=X*2+4:
    Y1=H*3+2: S1=S*2: O1=1
210 IF O=1 THEN 280
230 NEXT
240 PRINT@H*64,CHR$(30);: IF O1=0 THEN M=M+1:
    PRINT@38,M;
245 IF M=3 THEN 490
250 H=RND(3): D=D+SGN(.5-D)
260 IF O=0 THEN O1=0
270 GOTO 160
280 X=X+S: X2=X1: Y2=Y1: RESET(X1,Y1)
285 X1=X1+S1
290 G=G+.2: IF G<=0 THEN O=0: G=0: GOTO 230
300 Y1=Y1+G: IF X1<0 OR X1>127 THEN RESET
    (X2,Y2): O=0: G=0: O1=0: GOTO 420
310 F=15360+INT(Y1/3)*64+INT(X1/2): IF PEEK
    (F)=191 THEN O=0: G=0: GOTO 390
320 IF PEEK(F)=188 THEN PRINT@INT(Y1/3)*64+
    INT(X1/2)-1,"   ";: F1=F1+1: GOTO 340
330 SET(X1,Y1): GOTO 220
340 S1=S1/2: K=INT((X1-2)/6): IF I=0 THEN 360
350 IF K(I-1)=K THEN 370
360 K(I)=K: I=I+1
370 G=G-1
380 GOTO 330
390 PRINT@H*64,CHR$(30);: FOR Z=0 TO 1: P=0
400 FOR Z1=1 TO 10: F2=15553+Z1*64+3*K(Z)
410 IF PEEK(F2)=188 THEN P=P+1
420 NEXT Z1: FOR Z1=10 TO 1 STEP -1: F2=15553+
    Z1*64+3*K(Z): IF PEEK(F2)=191 THEN 450
430 IF P>0 THEN P=P+1: POKE F2,188: POKE F2+1,
    188: GOTO 450
440 POKE F2,32: POKE F2+1,32
450 NEXT Z1
460 NEXT Z: I=0
470 IF F1=0 THEN M=M+1: PRINT@38,M;: IF M=3
    THEN 490
480 HI=HI+F1: PRINT@6,HI;: F1=0: C$=INKEY$:
    O1=1: GOTO 240
490 PRINT@0,"HAI FATTO 3 ERRORI E"HI"CENTRI";
492 PRINT CHR$(30)
495 PRINT"PREMI ENTER PER GIOCARE DI NUOVO";:
    INPUT A: RUN
500 FOR X=256 TO 960 STEP 64: READ Y: PRINT@X
    ,STRING$(Y,191);: NEXT
505 DATA 21,18,15,12,9,9,9,15,15,15,63,63
510 PRINT@738,STRING$(3,191);: FOR X=802 TO
    866 STEP 64: PRINT@X,STRING$(8,191): NEXT
520 FOR X=299 TO 567 STEP 67: READ Y: PRINT@X
    ,STRING$(Y,191);: NEXT
525 DATA 20,17,14,11,8
530 FOR X=631 TO 887 STEP 64: PRINT@X,STRING$
    (8,191);: NEXT
540 GOTO 130
550 FOR X=1 TO 17: READ Y: NEXT: FOR X=384 TO
    960 STEP 64: READ Y: PRINT@X,STRING$(Y,191)
    ;: NEXT
555 DATA 9,9,12,12,15,15,15,15,63,63
```

```
560 FOR X=567 TO 931 STEP 64: PRINT@X,STRING$
    (8,191);: NEXT: PRINT@506,STRING$(5,191);:
    FOR X=278 TO 546 STEP 67: READ Y: PRINT@X,
    STRING$(Y,191);: NEXT
565 DATA 20,17,11,5,2
570 FOR X=610 TO 866 STEP 64: PRINT@X,STRING$
    (2,191);: NEXT: GOTO 130
600 FOR X=1 TO 32: READ Y: NEXT: FOR X=256 TO
    960 STEP 64: READ Y: PRINT@X,STRING$(Y,191)
    ;: NEXT
605 DATA 9,9,9,9,9,9,6,6,3,3.63.63
610 FOR X=1 TO 7: READ Y,Y1: PRINT@Y1,STRING$
    (Y,191);: NEXT
615 DATA 5,781,11,842,8.735,14,796,38,857.14.
    753,17,814
620 FOR X=506 TO 698 STEP 64: PRINT@X,STRING$
    (5,191);: NEXT: FOR X=1 TO 3: PRINT@X*67+
    238,STRING$((4-X)*3+5,191);: NEXT: GOTO 1[illegible]
```


franco muzzio editore

32 programmi con l'Apple

Il volume contiene 32 programmi in Basic completamente documentati con listati, esecuzioni di prova, istruzioni per fare girare il programma, suggerimenti per variazioni ecc. Tutti i programmi sono stati verificati e possono essere eseguiti su ogni tipo di Apple II L 9 500

Imparate il linguaggio dell'Apple

Se conoscete già il Basic e possedete un computer Apple siete sulla strada giusta per diventare un esperto in materia. Questo libro vi mette in grado di comprendere il linguaggio macchina dell'Apple partendo dagli esempi più semplici ed arrivando all'uso del mini assembler L 15 000

Il piacere del computer *è la prima collana interamente dedicata alle applicazioni hobbystiche e professionali del personal computer. Questi libri descrivono l'hardware e il software, insegnano la programmazione in vari linguaggi, offrono molteplici applicazioni e informazioni pratiche. Per conoscere gli altri titoli finora apparsi (relativi al PET/CBM, all'Apple, al Basic, al Pascal, al TRS-80 e ad altri argomenti) chiedete il catalogo generale a*

franco muzzio & c. editore
via bonporti 36 - 35100 padova

cognome e nome

indirizzo

cap località

PS2

Labirinto mobile

Lo scopo del gioco è spostare il punto lampeggiante attraverso le tre sezioni che costituiscono un labirinto in continuo movimento, dalla cima al fondo dello schermo. Ogni volta che il punto si scontra con un muro, in qualsiasi direzione, il computer scrive OOPS e il gioco ricomincia dall'inizio della divisione che avete urtato.

Il tempo è essenziale: dovete misurarvi con un orologio che scandisce i secondi di gioco in un angolo del video.

Avete a disposizione tre livelli di difficoltà: principiante, progredito e super-esperto (tra i nostri conoscenti non abbiamo ancora trovato nessun super-esperto).

Tandy
Radio Shack

```
1 REM ********************************************
2 REM *                                            *
3 REM *             LABIRINTO  MOBILE              *
4 REM *          VERSIONE TRS-80 MOD. I            *
5 REM *                                            *
6 REM *             PERSONAL SOFTWARE              *
7 REM *                                            *
8 REM ********************************************
9 REM
10 CLS: RANDOM: CLEAR 1500: DEFINT A-F,X-Z
12 PRINT@20,"LABIRINTO MOBILE"
15 GOSUB 210: GOSUB 20: GOSUB 290: GOTO 380
20 M$=STRING$(211,0)
22 R$=CHR$(26)+STRING$(5,24)
24 Q$=CHR$(188)+STRING$(3,131)+CHR$(188)+R$+
   CHR$(191)+STRING$(3,128)+CHR$(191)+R$+CHR$(143)
   +STRING$(3,176)+CHR$(143)
26 P$=CHR$(191)+STRING$(3,131)+CHR$(188)+R$+
   CHR$(191 +STRING$(3,140)+CHR$(131)+R$+
   CHR$(191)+STRING$(4,128)
28 S$=CHR$(188)+STRING$(3,131)+CHR$(140)+R$+
   CHR$(131)+STRING$(3,140)+CHR$(176)+R$+
   CHR$(140)+STRING$(3,176)+CHR$(143)
30 I=VARPTR(M$): A=0: B=1: C=15168: X-0: Y=0
31 Z=0: D=127: E=20: F=20: U=.33
35 J=PEEK(I+1)+256*PEEK(I+2)
39 IF J 32767 THEN J=-(65536-J)
40 FOR K=J TO J+210: READ X: POKE K,X: NEXT
50 DATA 58,127,60,245,33,126,60,17,127,60,1,63,0
52 DATA 237,184,58,128,60,50,64,60,33,129,60,17
60 DATA 128,60,1,63,0,237,176,58,255,60,50,191
70 DATA 60,33,254,60,17,255,60,1,63,0,237,184,58
80 DATA 0,61,50,192,60,33,1,61,17,0,61,1
90 DATA 63,0,237,176,241,50,63,61,58,191,61,245
100 DATA 33,190,61,17,191,61,1,63,0,237,184,58
110 DATA 192,61,50,128,61,33,193,61,17,192,61,1
120 DATA 63,0,237,176,58,63,62,50,255,61,33,62,62
130 DATA 17,63,62,1,63,0,237,184,58,64,62,50,0,62
140 DATA 33,65,62,17,64,62,1,63,0,237,176,241,50
150 DATA 127,62,58,255,62,245,33,254,62,17,255,62
155 DATA 1,63,0,237,184,58,0,63,50,192,62,33,1,63
160 DATA 17,0,63,1,63,0,237,176,58,127,63,50,63
165 DATA 63,33,126,63,17,127,63,1,63,0,237,184,58
170 DATA 128,63,50,64,63,33,129,63,17,128,63,1,63
175 DATA 0,237,176,241,50,191,63,201
190 POKE 16526,PEEK(I+1): POKE 16527,PEEK(I+2)
195 REM 190 DEFUSR0=J PER DISK-BASIC
196 RETURN
210 PRINT@64,"IL VOSTRO COMPITO E' SPOSTARE ";
215 PRINT"IL PUNTO LAMPEGGIANTE ATTRAVERSO"
220 PRINT"IL LABIRINTO MOBILE VERSO IL FONDO";
225 PRINT" DELLO SCHERMO.": PRINT
227 PRINT"TENENDO PREMUTI I TASTI A FRECCIA ";
230 PRINT"(UNO ALLA VOLTA) IL PUNTO SI"
235 PRINT"SPOSTERA' NELLA DIREZIONE INDICATA: ";
240 PRINT"SU', GIU', DESTRA, SINISTRA."
250 PRINT"CI SARANNO TRE DIVISIONI NEL ";
255 PRINT"LABIRINTO.  SE URTATE UN MURO,"
260 PRINT"DOVRETE RIPARTIRE DALL'INIZIO ";
265 PRINT"DELLA DIVISIONE CHE AVETE URTATO  '
270 PRINT"CERCATE DI PASSARE NEL MINOR TEMPO ";
275 PRINT"POSSIBILE. IL VOSTRO TEMPO E'";
280 PRINT"INDICATO NELL'ANGOLO SUPERIORE ";
285 PRINT"SINISTRO."
287 PRINT@850,"PAUSA...": RETURN
290 A$(1)=STRING$(5,140)+STRING$(6,128)+STRING$
    (5,140)
300 A$(2)=CHR$(191)+STRING$(5,128)+STRING$
    (4,140)+STRING$(5,128)+CHR$(191)
310 A$(3)=CHR$(143)+STRING$(4,131)+STRING$
    (6,176)+STRING$(4,131)+CHR$(143)
320 A$(4)=CHR$(191)+STRING$(4,128): A$(4)=A$(4)+
     A$(4)+A$(4)+CHR$(191)
330 A$(5)-CHR$(191)+STRING$(4,176)+STRING$
    (6,179)+STRING$(4,131)+CHR$(191)
340 A$(6)=CHR$(191)+STRING$(4,176)+CHR$(179)
    +STRING$(4,131)+CHR$(179)+STRING$(4,176)+
    CHR$(191)
350 A$(7)=CHR$(188)+CHR$(176)+CHR$(128)+CHR$
    (131)+CHR$(143)+CHR$(128)+STRING$(4,176)+
    CHR$(128)+CHR$(143)+CHR$(131)+CHR$(128)+
    CHR$(176)+CHR$(188)
360 A$(8)=CHR$(191)+STRING$(4,176)+CHR$(188)+
    CHR$(140)+STRING$(2,143)+CHR$(140)+CHR$
    (188)+STRING$(4,176)+CHR$(191)
370 A$(9)=CHR$(191)+STRING$(6,140)+CHR$(143)+
    CHR$(140)+CHR$(188)+STRING$(5,140)+CHR$
    (191): RETURN
380 PRINT@850,"BATTI =ENTER= PER GIOCARE":
    FOR I=1 TO 40: X$=INKEY$: IF X$<>""
    THEN 410 ELSE NEXT
382 PRINT@856,"          ";: FOR I=1 TO 20:
    X$=INKEY$: IF X$<>"" THEN 410 ELSE NEXT:
    GOTO 380
410 CLS: PRINT@266,"1=PRINCIPIANTE"
411 PRINT@330,"2=PROGREDITO"
415 PRINT@394,"3=SUPER-ESPERTO"
420 W=0: PRINT
422 INPUT"            CHE LIVELLO VUOI";S
425 IF S<0 OR S 3 THEN 420
430 CLS: PRINT@198,"PAUSA...": I=0: Y=1: A$="":
    B$="": C$="": ON S GOTO 440,470,500
440 FOR I=1 TO 15: A$=A$+A$(RND(4)): NEXT: A$=A$
    +STRING$(15,128)
450 FOR I=1 TO 15: B$=B$+A$(RND(4)): NEXT: B$-B$
    +LEFT$(A$(RND(3)+4),15)
460 FOR I=1 TO 15: C$=C$+A$(RND(4)): NEXT: C$=C$
    +LEFT$(A$(RND(2)+7),15): GOTO 600
470 FOR I=1 TO 15: A$=A$+A$(RND(7)): NEXT: A$=A$
    +LEFT$(A$(RND(3)+4),15)
```

```
480 FOR I=1 TO 15: B$=B$+A$(RND(7)): NEXT: B$=B$
    +LEFT$(A$(RND(2)+7),15)
490 FOR I=1 TO 15: C$=C$+A$(RND(6)+1): NEXT: C$=
    C$+LEFT$(A$(RND(2)+7),15): GOTO 600
500 FOR I=1 TO 15: A$=A$+A$(RND(6)+2): NEXT: A$=
    A$+LEFT$(A$(RND(4)+5),15)
510 FOR I=1 TO 15: B$=B$+A$(RND(6)+3): NEXT: B$=
    B$+LEFT$(A$(RND(2)+7),15)
520 FOR I=1 TO 15: C$=C$+A$(RND(5)+4): NEXT: C$=
    C$+LEFT$(A$(RND(3)+4),15)
600 CLS: PRINT@64,A$;: PRINT@384,B$;: PRINT@704,
    C$;
610 X=RND(80)+20: IF Y>29 THEN Y=31: SET(X,Y):
    GOTO 700
620 IF Y>14 THEN Y=16: SET(X,Y): GOTO 700
630 Y=1: SET(X,Y): GOTO 700
700 PRINT@0,INT(T);: RESET(X,Y): Z=PEEK(C): IF
    Z<>A GOSUB 900 ELSE SET(X,Y): FOR Q=1 TO E:
    NEXT
710 ON ERROR GOTO 1120: RESET(X,Y): L=USR(0):
    ON ERROR GOTO 0
720 IF POINT(X,Y) THEN 800 ELSE SET(X,Y)
730 RESET(X,Y): Z=PEEK(C): IF Z<>A GOSUB 900
    ELSE SET(X,Y): FOR Q=1 TO E: NEXT
740 T=T+U: RESET(X,Y): Z=PEEK(C): IF Z<>A GOSUB
    900 ELSE SET(X,Y): FOR Q=1 TO E: NEXT
745 RESET(X,Y):Z=PEEK(C): IF Z<>A GOSUB 900
    ELSE SET(X,Y): FOR Q=1 TO E: NEXT
750 GOTO 700
800 W=W+1: FOR I=1 TO 150: NEXT: CLS: PRINTCHR$
    (23): PRINT@200,O$+"  "+O$+"  "+P$+" "+S$;
810 FOR I=1 TO 800: NEXT: GOTO 600
900 FOR Q=1 TO F: IF Z=16 THEN 950
910 IF Z=32 THEN 970
920 IF Z=64 THEN 990
930 IF Z=8 THEN 1010
940 SET(X,Y): RETURN
950 Y=Y+B: IF Y=44 THEN 1100
960 IF POINT(X,Y) THEN 800 ELSE SET(X,Y): RETURN
970 X=X-B: IF X<B THEN X=B
980 IF POINT(X,Y) THEN 800 ELSE RESET(X,Y):
    RETURN
990 X=X+B: IF X>D THEN X=D
1000 IF POINT(X,Y) THEN 800 ELSE SET(X,Y): RETURN
1010 Y=Y-B: IF Y<B THEN Y=B
1020 IF POINT(X,Y) THEN 800 ELSE SET(X,Y): RETURN
1100 PRINT@0,"      ";: PRINT@960,FIX((34000-T*10
     -5*W)*S);"PUNTI :";INT(T);" SECONDI (E";W;
     " OOPS).";
1105 PRINT"  PREMI =ENTER=";
1110 INPUTX$: CLS: GOTO 410
1120 CLS: PRINT"DEVI SOSTITUIRE L'ATTUALE RIGA 190"
1125 PRINT"PER USARE QUESTO PROGRAMMA CON IL DISK-";
1130 PRINT"BASIC": LIST 190-195
```

Frasi inutili

Prendendo spunto dal gergo usato dai giornali, questo programma scrive frasi che sembrano effettivamente tratte da discorso di qualche politico nostrano. Dopo ogni composizione il computer vi chiede se aveste ancora voglia di leggerne un'altra oppure se preferite che smetta.

commodore

```
1 REM ***********************************************************************
2 REM *                                                                     *
3 REM *                          FRASI INUTILI                              *
4 REM *                                                                     *
5 REM *                  VERSIONE PET/CBM [illegible]                       *
6 REM *                                                                     *
7 REM *                       [illegible]                                   *
8 REM *                                                                     *
9 REM ***********************************************************************
10 CLR
15 DIM A$(10),B$(10),C$(10),D$(10),E$(10),[illegible]
20 PRINT CHR$(147)
30 PRINT: PRINT: PRINT: PRINT CHR$(1[illegible]"GENERATORE [illegible]"
40 PRINT: PRINT: PRINT"DA UNO STUDIO EFFETTUATO [illegible]
50 PRINT"MARCHI DELL'UNIVERSITA' DI PISA, [illegible]
60 PRINT"PROF. PIERO MORO[illegible]
70 PRINT"RATORIO PRESSO L'ISTITUTO SUPERIORE DI
80 PRINT"SANITA  "
90 GOSUB [illegible]
100 PRINT"[illegible]
110 PRINT"TARIA E DA ARTICOLI [illegible]
120 PRINT"[illegible]
130 PRINT"[illegible]
140 PRINT"[illegible]
150 PRINT: PRINT"COMBINANDOLE [illegible]
160 PRINT"TUTTI I MODI POSSIBILI, [illegible]
170 PRINT"MARE OLTRE [illegible]
180 PRINT"TI SENSO COMPIUTO (SI E) [illegible]
```

```
190 PRINT"CHE POTREBBERO BENISSIMO FAR PARTE DI"
200 PRINT"ULTERIORI ARTICOLI SULL'ARGOMENTO."
210 GOSUB 10000
220 PRINT CHR$(147)
225 PRINT: PRINT: PRINT: PRINT: PRINT: PRINT
230 PRINT TAB(12) CHR$(18)"STO PENSANDO!"
240 FOR K=1 TO 2000: NEXT K
250 PRINT CHR$(147)
1010 A$(1)="L'UTENZA POTENZIALE": A$(2)="IL BISOGNO EMERGENTE"
1020 A$(3)="IL QUADRO NORMATIVO": A$(4)="LA VALENZA EPIDEMIOLOGICA"
1030 A$(5)="IL NUOVO SOGGETTO SOCIALE": A$(6)="L'APPROCCIO PROGRAMMATICO"
1040 A$(7)="L'ASSETTO POLITICO ISTITUZIONALE": A$(8)="IL CRITERIO METODOLOGICO"
1050 A$(9)="IL MODELLO DI SVILUPPO": A$(10)="IL METODO PARTECIPATIVO"
1060 B$(1)="SI CARATTERIZZA PER": B$(2)="PRIVILEGIA": B$(3)="PREFIGURA"
1070 B$(4)="RICONDUCE A SINTESI": B$(5)="PERSEGUE": B$(6)="ESTRINSECA"
1080 B$(7)="SI PROPONE": B$(8)="PRESUPPONE": B$(9)="PORTA AVANTI"
1090 B$(10)="AUSPICA"
1100 C$(1)="IL RIBALTAMENTO DELLA LOGICA ASSISTENZI-ALE PREESISTENTE"
1110 C$(2)="IL SUPERAMENTO DI OGNI OSTACOLO E/O RE- SISTENZA PASSIVA"
1120 C$(3)="UN ORGANICO COLLEGAMENTO INTERDISCIPLI- NARE ED UNA PRASSI DI"
1130 C$(3)=C$(3)+" LAVORO DI GRUPPO"
1140 C$(4)="LA PUNTUALE CORRISPONDENZA FRA OBIETTI- VI E RISORSE"
1150 C$(5)="LA VERIFICA CRITICA DEGLI OBIETTIVI I-  STITUZIONALI E L'INDI"
1160 C$(5)=C$(5)+"VIDUAZIONE DI FINI QUALIFICANTI"
1170 C$(6)="IL RIORIENTAMENTO DELLE LINEE DI TEN-   DENZA IN ATTO"
1180 C$(7)="L'ACCORPAMENTO DELLE FUNZIONI ED IL DE- CENTRAMENTO DECISIONALE"
1190 C$(8)="LA RICOGNIZIONE DEL BISOGNO EMERGENTE E DELLA DOMANDA NON "
1200 C$(8)=C$(8)+"SODDISFATTA"
1210 C$(9)="LA RICONVERSIONE ED ARTICOLAZIONE PERI- FERICA DEI SERVIZI"
1220 C$(10)="UN CORRETTO RAPPORTO FRA STRUTTURE E SO-VRASTRUTTURE"
1230 D$(1)="NEL PRIMARIO INTERESSE DELLA POPOLAZIONE"
1240 D$(2)="SENZA PREGIUDICARE L'ATTUALE LIVELLO    DELLE PRESTAZIONI"
1250 D$(3)="AL DI SOPRA DI INTERESSI E PRESSIONI DI PARTE"
1260 D$(4)="SECONDO UN MODULO DI INTERDIPENDENZA    ORIZZONTALE"
1270 D$(5)="IN UNA VISIONE ORGANICA E RICONDOTTA A  UNITA'"
1280 D$(6)="CON CRITERI NON DIRIGISTICI"
1290 D$(7)="AL DI LA' DELLE CONTRADDIZIONI E DIFFI- COLTA' INIZIALI"
1300 D$(8)="IN MANIERA ARTICOLATA E NON TOTALIZZANTE"
1310 D$(9)="ATTRAVERSO I MECCANISMI DELLA PARTECI-  PAZIONE"
1320 D$(10)="SENZA PRECOSTITUZIONE DELLE RISPOSTE"
1330 E$(1)="SOSTANZIANDO E VITALIZZANDO"
1340 E$(2)="RECUPERANDO OVVERO RIVALUTANDO"
1350 E$(3)="IPOTIZZANDO E PERSEGUENDO"
1360 E$(4)="NON ASSUMENDO MAI COME IMPLICITO"
1370 E$(5)="FATTUALIZZANDO E CONCRETIZZANDO"
1380 E$(6)="NON SOTTACENDO MA ANZI PUNTUALIZZANDO"
1390 E$(7)="POTENZIANDO ED INCREMENTANDO"
1400 E$(8)="NON DANDO CERTO PER SCONTATO"
1410 E$(9)="EVIDENZIANDO ED ESPLICITANDO"
1420 E$(10)="ATTIVANDO ED IMPLEMENTANDO"
1430 F$(1)="NEI TEMPI BREVI, ANZI BREVISSIMI"
1440 F$(2)="IN UN'OTTICA PREVENTIVA E NON PIU' CURA-TIVA"
1450 F$(3)="IN UN AMBITO TERRITORIALE OMOGENEO,AI   DIVERSI LIVELLI"
1460 F$(4)="NEL RISPETTO DELLA NORMATIVA ESISTENTE"
1470 F$(5)="NEL CONTESTO DI UN SISTEMA INTEGRATO"
1480 F$(6)="QUALE SUA PREMESSA INDISPENSABILE E CON-DIZIONANTE"
1490 F$(7)="NELLA MISURA IN CUI CIO' SIA FATTIBILE"
1500 F$(8)="CON LE DOVUTE ED IMPRESCINDIBILI SOTTO- LINEATURE"
1510 F$(9)="IN TERMINI DI EFFICACIA E DI EFFICIENZA"
1520 F$(10)="A MONTE E A VALLE DELLA SITUAZIONE CON- TINGENTE"
1530 G$(1)="LA TRASPARENZA DI OGNI ATTO DECISIONALE"
1540 G$(2)="LA NON SANITARIZZAZIONE DELLE RISPOSTE"
1550 G$(3)="UN INDISPENSABILE SALTO DI QUALITA'"
1560 G$(4)="UNA CONGRUA FLESSIBILITA' DELLE STRUT-  TURE"
1570 G$(5)="L'ANNULLAMENTO DI OGNI GHETTIZZAZIONE"
1580 G$(6)="IL COINVOLGIMENTO ATTIVO DI OPERATORI ED UTENTI"
1590 G$(7)="L'APPIANAMENTO DELLE DISCREPANZE E DELLE DISCRASIE ESISTENTI"
1600 G$(8)="LA RIDEFINIZIONE DI UNA NUOVA FIGURA    PROFESSIONALE"
1610 G$(9)="L'ADOZIONE DI UNA METODOLOGIA DIFFEREN- ZIATA"
1620 G$(10)="LA DEMEDICALIZZAZIONE DEL LINGUAGGIO"
2000 A1=INT(10*RND(1)+1): B1=INT(10*RND(1)+1)
2010 C1=INT(10*RND(1)+1): D1=INT(10*RND(1)+1)
2020 E1=INT(10*RND(1)+1): F1=INT(10*RND(1)+1)
2030 G1=INT(10*RND(1)+1)
2500 PRINT A$(A1)
2510 PRINT B$(B1)
2520 PRINT C$(C1)
2530 PRINT D$(D1)
2540 PRINT E$(E1)
2550 PRINT F$(F1)
2560 PRINT G$(G1)
3000 PRINT: PRINT: PRINT CHR$(18)"VUOI CONTINUARE"
3005 GET W$: IF W$="" THEN 3005
3010 IF LEFT$(W$,1)="S" THEN 220
3020 PRINT: PRINT"OK - SPERO DI ESSERTI STATO D'AIUTO!!"
3030 END
10000 PRINT: PRINT: PRINT TAB(10) CHR$(18)"(PREMI UN TASTO)"
10010 GET A$: IF A$="" THEN 10010
10020 PRINT CHR$(147): RETURN
```

Julia

Il programma stampa un calendario giuliano per l'anno richiesto.

Variabili principali

M1$, M2$ e M3$ sono stringhe che contengono i giorni dei mesi;
L$ contiene la lunghezza dei mesi;
M$ contiene il nome dei 12 mesi;
C$ contiene i numeri corrispondenti ai giorni del mese;
FNF(Y) è la funzione per trovare il primo giorno dell'anno;
FNM(V.1,V2) è la funzione che calcola il modulo di un numero;
D$ contiene i giorni della settimana;
Y è l'anno di cui si desidera il calendario.

```
                              **********
                              *  1982  *
                              **********

         GENNAIO                      FEBBRAIO                      MARZO

lu ma me gi ve sa do         lu ma me gi ve sa do         lu ma me gi ve sa do

             1  2  3          1  2  3  4  5  6  7          1  2  3  4  5  6  7
 4  5  6  7  8  9 10          8  9 10 11 12 13 14          8  9 10 11 12 13 14
11 12 13 14 15 16 17         15 16 17 18 19 20 21         15 16 17 18 19 20 21
18 19 20 21 22 23 24         22 23 24 25 26 27 28         22 23 24 25 26 27 28
25 26 27 28 29 30 31                                      29 30 31

         APRILE                       MAGGIO                        GIUGNO

lu ma me gi ve sa do         lu ma me gi ve sa do         lu ma me gi ve sa do

          1  2  3  4                         1  2             1  2  3  4  5  6
 5  6  7  8  9 10 11          3  4  5  6  7  8  9          7  8  9 10 11 12 13
12 13 14 15 16 17 18         10 11 12 13 14 15 16         14 15 16 17 18 19 20
19 20 21 22 23 24 25         17 18 19 20 21 22 23         21 22 23 24 25 26 27
26 27 28 29 30               24 25 26 27 28 29 30         28 29 30
                             31

         LUGLIO                       AGOSTO                        SETTEMBRE

lu ma me gi ve sa do         lu ma me gi ve sa do         lu ma me gi ve sa do

          1  2  3  4                            1               1  2  3  4  5
 5  6  7  8  9 10 11          2  3  4  5  6  7  8          6  7  8  9 10 11 12
12 13 14 15 16 17 18          9 10 11 12 13 14 15         13 14 15 16 17 18 19
19 20 21 22 23 24 25         16 17 18 19 20 21 22         20 21 22 23 24 25 26
26 27 28 29 30 31            23 24 25 26 27 28 29         27 28 29 30
                             30 31

         OTTOBRE                      NOVEMBRE                      DICEMBRE

lu ma me gi ve sa do         lu ma me gi ve sa do         lu ma me gi ve sa do

             1  2  3          1  2  3  4  5  6  7                1  2  3  4  5
 4  5  6  7  8  9 10          8  9 10 11 12 13 14          6  7  8  9 10 11 12
11 12 13 14 15 16 17         15 16 17 18 19 20 21         13 14 15 16 17 18 19
18 19 20 21 22 23 24         22 23 24 25 26 27 28         20 21 22 23 24 25 26
25 26 27 28 29 30 31         29 30                        27 28 29 30 31
```

Star War

Si tratta di un gioco di tipo spaziale consistente nel distruggere una navicella aliena che si muove in modo casuale nella parte alta dello schermo. Bisogna però fare molta attenzione perché tra la navicella da voi pilotata e l'astronave aliena vi è una barriera e tutte le volte che la colpirete, l'astronave aliena si fermerà e lascerà cadere una bomba, che vi distruggerà solo se colpirà le ali della vostra astronave.
L'astronave aliena può sparare solo in due modi: (a) in modo verticale, e (b) in modo obliquo. Per proteggervi potrete usufruire di una barrirea che neutralizza i proiettili alieni, solo però quelli che cadono in modo verticale. La barriera può essere usata solo dieci volte.
I tasti per il movimento della vostra astronave sono < per andare a sinistra e > per andare a destra, per azionare la barriera basta premere C mentre per sparare Z. Si noterà che quando si premono i pulsanti per lo spostamento della navicella vi è un rallentamento del gioco. Ciò è dovuto al fatto che il VIC non è in grado di gestire video e audio contemporaneamente e che il Basic non è così veloce da permettere la realizzazione di giochi troppo elaborati.

Principali routine

100	Cambia il colore allo schermo e richiama la subroutine di presentazione
110-130	Inizializza le variabili e dà il repeat a tutti i tasti
140	Alza il volume dei generatori audio
160	Genera la barriera
170-200	Genera l'astronave
210-240	Genera la navicella aliena
250-270	Cancella l'astronave
290-340	Controlla i comandi dell'astronave
350-370	Genera nella nuova posizione l'astronave
400-420	Cancella la navicella aliena
420-430	Genera il motivo audio
440-470	Muove la navicella aliena
480-500	Controlla se la vostra navicella è sotto tiro della navicella aliena
520-530	Come 210-240
540-620	Fa sparare la navicella aliena e controlla se vi ha colpito
630-640	Controlla se il proiettile da voi sparato ha colpito la barriera
650-670	Controlla se il proiettile da voi sparato ha colpito l'astronave aliena
700-770	Genera bomba
780-820	Genera proiettili alieni
830-870	Genera esplosione audio e video
890-900	Fa esplodere la navicella aliena
910	Fa esplodere la vostra navicella
920-1120	Presentazione del gioco

```
100 POKE36879,8 GOSUB920
110 CLR DATA125 187 179 175,0 READ[illegible]
120 POKE650 128
130 J=4 P=7680 G=36877 M=7680+22 [illegible] E=10 [illegible]
140 POKE36878 15
150 PRINT [illegible]
160 [illegible]
170 PRINT [illegible]
180 PRINT [illegible]
190 PRINT [illegible]
200 PRINT [illegible]
210 GETA$ [illegible]
220 [illegible]
230 [illegible]
240 [illegible] 32
250 PRINTTAB[illegible]
260 PRINTTAB[illegible]
270 [illegible]
280 [illegible]
290 [illegible]
300 IFA$= [illegible] THEN[illegible]
310 IFA$= [illegible]
320 [illegible]
330 [illegible]
340 IF[illegible]THEN[illegible]
350 PRINTTAB[illegible]
360 [illegible]
370 [illegible]
[illegible]
```

```
750 POKEN+(U-22),80
760 FORT=0TO100:NEXTT
770 NEXT:POKE36875,0:P=0:O=0:RETURN
780 FORK=0TO3:FORA=I-1TOI+5
790 IFPEEK(PP+A)=160THENU=0
800 POKEPP+A,160:NEXT:PP=PP+22:NEXT
810 PP=7875-44:FORK=0TO3:FORA=I-1TOI+5
820 POKEPP+A,32:NEXTA:PP=PP+22:NEXT:POKE36[illegible]:RETURN
830 POKE3 220 [illegible]
840 IFTP=1THEN[illegible]
850 FORL=15TO0STEP-1
860 POKE36878,L:FORT=1TO300:NEXTT:NEXT
870 POKEG,0:POKE36878,0:POKE36879,5
880 IFTP=1THENRUN
890 PRINT"[illegible]":PRINTTAB(U)"/——":PRINTTAB(U)"/ |":PRINTTAB(U)"\OOO/":FORT=0TO3
900 NEXT:CLR:GOTO110
910 FORT=0TO2000:NEXTT:GOTO130
910 PRINTTAB(I+1)"/":PRINTTAB(I+1)"- -":PRINTTAB(I+1)"/\":RETURN
920 PRINT"[illegible]",
930 PRINT"[illegible]"
940 PRINT"[illegible]"
950 PRINT"[illegible]"
960 PRINT"[illegible]"
970 PRINT"[illegible]"
980 PRINT"[illegible]"
990 PRINT"[illegible]"
1000 PRINT"[illegible]"
1010 PRINT"[illegible]"
1020 PRINT"[illegible]"
1030 PRINT"[illegible]"
1040 PRINT"[illegible]"
1050 PRINT"[illegible]"
1060 PRINT"[illegible]"
1070 PRINT"[illegible]"
1080 PRINT"[illegible]BY FELICE GRANITO"
1090 D=INT(RND(1)*7)+1:FORA=0TO252:POKE38400+A,D:POKE38905-A,D-1
1100 GETA$:IFA$=""THEN1120
1110 GOTO120
1120 NEXT:GOTO1090
```

Crazy car

Il programma qui presentato è l'elaborazione di un videogioco molto diffuso: l'automobilista pazzo. Il gioco consiste nel far saltare ad una auto (con relativo pilota) una serie di trampolini, i quali, man mano che il gioco prosegue, vanno gradualmente ingrandendosi, aumentando così la difficoltà del gioco. Il trampolino è costituito da una pedana di salto di una serie di bidoni che possono raggiungere un massimo di 19, e da una pedana di arrivo. L'abilità del pilota consiste nell'aumentare o nel diminuire la velocità dell'auto secondo la lunghezza del trampolino. I tasti per l'accelerazione sono > e < ma possono essere facilmente cambiati modificando le linee 550 e 560. Appena si inizia a giocare, la velocità dell'auto è tale da permettere all'auto stessa di saltare un trampolino costituito da un solo bidone. Accelerando si sentirà che il rombo dell'auto sale di tono, ma attenzione perché gli ostacoli da saltare non sono tutti uguali. Infatti nel programma vi è una piccola routine (linee 290-340) che fa in modo che la lunghezza dei trampolini sia generata casualmente. Per riuscire a superare i trampolini bisogna cadere giusto dopo la pedana di arrivo, né un po' più avanti né un po' più indietro, se non si vuole correre il rischio di fare una brutta caduta.

Variabili usate

A$	stringa per comunicare al sistema i comandi del gioco
P(A)	coordinate di inizio dei trampolini
K(A)	coordinate di fine dei trampolini
KK	codici del colore
U e I	coordinate dell'auto
SC	punteggio della gara
HS	massimo punteggio
G e Z	usate per il calcolo della lunghezza del salto
O	posizione dell'auto durante il salto
A, M, P	variabili di ciclo
T	velocità del gioco

VIC 20

```
100 D=1:A=242:M=0:B=5
110 POKE36879,8:PRINT"[illegible]",
120 PRINT"[illegible]FELICE GRANITO GAMES[illegible]"
130 S$="[illegible]":KK$="[illegible]"
140 PRINT"[illegible]":PRINTLEFT$(S$,M)+LEFT$(KK$,X)"[illegible]CRAZY CAR"
150 POKE7680+A,97:POKE7680+(A+1),87:POKE7680+(A-22),81
160 FORT=0TO50:NEXTT
170 POKE7680+A,32:POKE7680+(A+1),32:POKE7680+(A-22),32
180 A=A-1:M=M+1:X=X+1
190 IFA<262THENA=242
200 IFM>8THENM=M+2
210 IFM>18THENM=18:B=B+1
220 IFB=[illegible]THENFORA=[illegible]2TO442:POKE7680+A,32:NEXT:A=242
230 IFX>7THENX=0
240 IFB=70THENGOTO100
250 GETA$:IFA$=""THENGOTO140
260 POKE36879,10
270 PRINT"[illegible]"
280 L=7856:C=30720:KK=1:U=L-22:T=100:Z=4
290 FORA=1TO3
300 P(A)=INT(RND(1)*8)+1
310 IFP(A)<3THEN300
320 K(A)=P(A)+INT(RND(1)*[illegible])
330 IFK(A)=K(A-1)GOTO300
340 NEXT
350 PRINT"[illegible]"
360 FORA=1TO4
370 PRINT"[illegible]————————————"
380 NEXT
390 FORP=1TO3
400 FORB=P(P)TOK(P)
410 POKEL+B,[illegible]:POKEC+L+B,KK:KK=KK+1
420 IFKK>5THENKK=1
```

```
430 NEXTB PRINT"FELICE GRANITO GAMES" .
440 POKEL+B,223 POKEL+[illegible](A)-1),233 [illegible](A)=2
450 [illegible]
460 NEXTA
470 REM [illegible]
480 FORJ=7724TO8144STEP110
490 FORI=0TO20 SC=SC+1
500 IFPEEK((U+22)+(I+1))=233THENGOSUB760
510 IFPEEK([illegible])=223ORPEEK((U+22)+(I+1))=223THEN920
520 PRINT"SCORE "SC PRINT"HI-SCORE "HS
530 [illegible] POKE(J+22)+I,87 POKE(J+22)+(I+1),87
540 GETA$ IFA$=""THEN570
550 IFA$="," THEN[illegible]
560 IFA$="."THEN[illegible]
570 [illegible]THEN[illegible]
580 IFT>100THENT=100
590 IFZ<4THENZ=4
600 IFZ>17THENZ=17
610 G=INT(I+Z)
620 GOSUB730 FORP=0TOT NEXTP
630 POKEL+I,32 POKE(J+22)+I,32 POKE(J+22)+(I+1),32
640 NEXTI NEXTJ
650 REM INCREMENTO OSTACOLO
660 GOSUB1190
670 [illegible]THEN[illegible]
680 FORA=1TO[illegible]
690 [illegible](A)+1
700 NEXTA
710 IFK(1)=19ORK(2)=19ORK(3)=19THENL=7856 GOTO[illegible]
720 L=7856 GOTO350
730 POKE36875,195+(Z*2)
740 POKE36874,200+(Z*2)
750 POKE36874,0 RETURN
760 REMSALTO
770 POKE(U-22)+I,81 POKEU+I,87 POKEU+(I+1),87
780 FORP=0TOT NEXTP
790 POKE(U-22)+I,32 POKEU+I,32 POKEU+(I+1),32
800 FORO=(I+1)TOG
810 POKE(J-22)+O,81 POKE(J-22)+(O+1),87 POKE(J-44)+(O+1),87
820 GOSUB730 FORP=0TOT NEXTP
830 POKE(J-22)+O,32 POKE(J-22)+(O+1),32 POKE(U-44)+(O+1),32 SC=SC+5
840 NEXT I=O
850 IFO<>ZZ(D)+1ORO>20THEN920
860 D=D+1
870 IFD>3THEND=1
880 IFPEEK(U+I)=233ORPEEK(U+I)=160ORPEEK((U+22)+I)=160THENGOSUB920
890 IFPEEK((J+22)+I)=233THENGOSUB920
900 RETURN
910 REM[illegible]
920 FORO=1TO19
930 POKE36874,0 POKE36877,241
940 POKE(U+22)+O,81 POKEU+O,87 POKE(J+22)+(O+1),87
950 FORT=0TO100 NEXTT
960 POKE(U+22)+O,32 POKEU+O,32 POKE(U+22)+(O+1),32
970 NEXTO
980 POKE36877,0
990 FORO=17TO0STEP-1
1000 POKE(J+22)+(O+2),81 POKE(U+22)+O,81 POKEU+(O+2),160 POKEU+(O+1),171 POKEU+O
,233
1010 POKEC+(U+(O+1)),KK KK=KK+1 IFKK>2THENKK=1
1020 GOSUB1050
1030 POKE(U+22)+(O+2),32 POKE(U+22)+O,32 POKEU+(O+2),32 POKEU+(O+1),32 POKEU+O,3
2
1040 NEXT GOTO1070
1050 POKE36875,200 FORM=1TO100 NEXTM POKE36875,0
1060 POKE36876,200 FORM=1TO100 NEXTM POKE36876,0 RETURN
1070 FORT=0TO200 NEXT
1080 IFSC>HSTHENHS=SC
1090 PRINT"" A$="" SC=0
1100 PRINT"[illegible] CONTINUARE(S/N)"
1110 GETA$ IFA$=""THEN1140
1120 IFA$="N"THEN100
1130 GOTO270
1140 FORA=176TO176+21
1150 POKEC+(7680+A),KK
1160 KK=KK+1 IFKK>6THENKK=0
1170 NEXT
1180 GOTO1110
1190 DATA135,159,175,183,187,183,175,159,135,195,135
1200 POKE36875,0 POKE36877,0
1210 FORAA=0TO10 READO POKE36874,O
1220 FORP=0TO150 NEXTP
1230 NEXT POKE36874,0 RESTORE RETURN
```

Motocross

Il programma simula una gara di motocross, o meglio il salto di ostacoli. Gli ostacoli sono costituiti da quadratini disposti per un certo numero, uno distante dall'altro, su quattro file: il numero di quadrati aumenta e lo spazio fra loro diminuisce man mano che si prosegue nel gioco. Si tratta di un gioco di tempismo: premendo il tasto S (salto) la moto si alza da terra e, se il tasto è stato premuto al momento giusto, supera l'ostacolo. Ad ogni salto verrà attribuito un determinato punteggio, il punteggio più alto verrà sempre visualizzato, mentre il punteggio attuale sarà azzerato ad ogni gara. Per mantenere fissa la difficoltà del gioco, basta cancellare le linee 410, 420, 430.

Principali routine

100	Cambia il colore allo schermo e richiama la subroutine di presentazione del gioco
110	Inizializza le variabili
130-150	Istruzioni
160-210	Genera il percorso e gli ostacoli
220-250	Spostamento della moto
260-310	Controlla se vi sono comandi e genera la figura della moto
320	Incrementa il punteggio
330-370	Cancella la vecchia posizione della moto
380-400	Genera il motivo audio del fine percorso
410-440	Incrementa gli ostacoli e aumenta la velocità del gioco
450-540	Salto della moto
560-660	Visualizza la moto capovolta e genera un motivo musicale
680	Controlla i punteggi
740-960	Presentazione

```
100 POKE[illegible] GOSUB[illegible]
110 [illegible]=30 [illegible]=1 [illegible]=1 SC=0 RESTORE
120 POKE[illegible] 10
130 PRINT [illegible]
140 [illegible] PREMI UN TASTO PRINT PER PARTIRE PRINT [illegible] PER SA
[illegible]
150 GETA$ [illegible]THEN[illegible]
160 PRINT [illegible]
170 FORA=1TO4 PRINT[illegible]
180 [illegible]
190 [illegible]STEP110 FORB=4TO16STEP[illegible]
200 POKEA+B,160 POKE(30720+(A+B)),6 NEXTB NEXTA
210 FORT=0TO1000 NEXT
220 [illegible]
230 [illegible]
240 [illegible],35
250 IF[illegible]2-INT(X/2[illegible]THEN[illegible]
```

```
[illegible]
```

Pontoon

Questo programma simula il gioco di carte Poonton come viene giocato dalle macchinette che si trovano alle fiere di paese. Le spiegazioni per giocare vengono date dal programma e, cosa molto importante, il computer non bara, cosicché avete il 50% di probabilità di vincere.

Il computer fa da banco e, all'inizio del gioco, avete un capitale di 1000 lire. In ogni caso, il banco vi fa credito assegnandovi, per esempio, L. — 1000. Per vincere dovete superare il punteggio delle carte del computer e, per fare più soldi, dovete tentare di fare "pontoon", ossia la combinazione di un ASSO con un DIECI o una figura (J, Q o K).

L'asso può assumere sia il valore 1 che 11 e il banco, una volta che ve ne sia uscito uno, vi chiede di decidere il valore che desiderate durante la mano di gioco. (N.B.: una volta scelto un valore non potete tornare sulla vostra decisione). Segue una lista di tutte le lettere che appaiono in carattere inverso e delle variabili usate nel programma.

Riga 610 - "(6 £ inverse) (*hai fatto pontoon* inverso) (5 £ inverse)"

Riga 620 - "(4 £ inverse) (*hai vinto con 5 carte* inverso)"

Riga 630 - "(9 £ inverse) (*hai vinto* inverso) (9 £ inverse)"

Riga 640 - "(9 £ inverse) (*hai perso* inverso) (9 £ inverse)"

Righe dalla 1015 alla 1050:

A inversa - D inversa
B inversa - J inversa
C inversa - Q inversa
D inversa - K inversa
E inversa - A inversa

Righe dalla 1070 alla 1100: le lettere inverse sono G, P, Q e F

Variabili usate:

Z	denaro disponibile
A$	separa le mani del gioco sullo schermo
B$	carte del I e del II giocatore
D	punteggio del giocatore
F	punteggio del computer
G	numero delle carte del giocatore
H	numero di assi persi per ogni mano dal giocatore
I	numero di carte del computer
C$	valore della carta
D$	seme della carta
E	usata per aggiungere carte sullo schermo
T	usata nelle istruzioni PRINT AT
Z	contiene il valore attribuito dal giocatore all'asso.

sinclair ZX81

```
   1 REM *** PONTOON ***
  10 PRINT TAB 9;"PONTOON"
  15 PRINT
  20 PRINT "******* IO SONO IL BANCO *******"
  30 PRINT
  32 PRINT "DEVI SUPERARE IL PUNTEGGIO"
  34 PRINT "DELLE MIE CARTE PER VINCERE."
  36 PRINT "QUESTE SONO LE VINCITE:"
  38 PRINT
  40 PRINT "SE SUPERI LE MIE CARTE-£1000"
  45 PRINT
  50 PRINT "SE HAI UN GIOCO DI 5 CARTE-£3000"
  55 PRINT
  60 PRINT "SE FAI ""PONTOON""-£5000"
  65 PRINT
  70 PRINT "PER GIOCARE OCCORRONO £1000."
  80 PRINT AT 20,0;"SE SEI PRONTO PREMI UN TASTO"
  90 IF INKEY$="" THEN GOTO 90
 100 LET Z=1000
 102 LET A$="[illegible]"
 103 LET B$="[illegible] "
 104 LET D=0
 105 LET F=0
 106 LET G=2
 107 LET H=0
 108 LET I=0
 109 CLS
 110 PRINT AT 0,0;"TU HAI £";Z
 120 PRINT A$
 130 PRINT AT 2,0;"********* LE MIE CARTE *********"
 140 FOR T=1 TO 7
 141 PRINT B$;B$,,
 142 NEXT T
 150 PRINT AT 10,0;A$
 160 PRINT "********* LE TUE CARTE *********"
 170 FOR T=1 TO 7
 171 PRINT B$;B$,,
 172 NEXT T
 180 PRINT A$
 181 PRINT AT 20,0;"PREMI UN TASTO PER GIRARE          LE TUE CARTE"
 182 IF INKEY$="" THEN GOTO 182
 185 PRINT AT 20,0;"                                "
 190 GOSUB 1000
 195 IF C=11 THEN LET H=1
 200 PRINT AT 12,0;C$,TAB 4;D$
 210 PRINT AT 18,0;D$;TAB 4;C$
 215 LET D=D+C
 220 GOSUB 1000
 225 IF C=11 THEN LET H=H+1
 230 PRINT AT 12,6;C$;TAB 10;D$
 240 PRINT AT 18,6;D$,TAB 10;C$
 250 LET D=D+C
 252 IF D=21 THEN GOTO 500
 253 IF H=1 OR H=2 THEN GOSUB 2000
 255 LET E=0
 260 PRINT AT 21,0;"(F) PER FERMARSI (C) PER CARTA"
 280 IF INKEY$="" THEN GOTO 280
 290 IF INKEY$="F" THEN GOTO 500
 292 IF INKEY$<>"C" THEN GOTO 280
 300 FOR T=12 TO 18
 310 PRINT AT T,12+E;B$
 320 NEXT T
 330 GOSUB 1000
 340 PRINT AT 12,12+E;C$;TAB 16+E;D$
 350 PRINT AT 18,12+E;D$;TAB 16+E;C$
 360 LET D=D+C
 365 IF C=11 THEN GOSUB 2000
 370 LET E=E+6
 375 LET G=G+1
 377 IF D>21 THEN GOTO 601
 378 IF E=18 THEN LET E=12
 380 GOTO 250
 500 LET E=0
 511 IF F=21 AND I=2 THEN GOTO 640
 512 IF D=21 AND I=2 THEN GOTO 606
 515 IF F>=D THEN GOTO 605
 520 FOR T=3 TO 9
 530 PRINT AT T,E;B$
 540 NEXT T
 550 GOSUB 1000
 555 IF C=11 THEN GOSUB 3000
 560 PRINT AT 3,E;C$;TAB E+4;D$
 570 PRINT AT 9,E;D$;TAB E+4;C$
 580 LET F=F+C
 590 LET E=E+6
 595 LET I=I+1
 596 IF E=30 THEN LET E=24
 600 GOTO 510
 601 LET D=-1
 605 IF F>21 THEN LET F=0
 606 PRINT AT 20,0;"                                "
 610 IF D=21 AND G=2 THEN PRINT AT 21,0;"£££££££££ HAI FATTO PONTOON ££££"
 620 IF D>F AND G=5 THEN PRINT AT 21,0;"£££££ HAI VINTO CON CINQUE CARTE"
 630 IF (D>F AND G<>2 AND G<>5) OR (D>F AND D<>21 AND G<>5) THEN PRINT AT 21,0;"£££££££££££££ HAI VINTO £££££££££"
 640 IF F>=D THEN PRINT AT 21,0;"£££££££££ HAI PERSO ££££££££££££££"
 650 IF D=21 AND G=2 THEN LET Z=Z+5000
 660 IF D>F AND G=5 THEN LET Z=Z+3000
 670 IF (D>F AND G<>2 AND G<>5) OR (D>F AND D<>21 AND G<>5) THEN LET Z=Z+1000
 680 IF F>=D THEN LET Z=Z-1000
 690 PAUSE 150
 700 GOTO 104
1000 LET A=INT (RND*12)+29
1001 LET C=A-28
1002 IF C>10 THEN LET C=10
1003 IF A=29 THEN LET C=11
1010 LET C$=CHR$ (A+128)
1015 IF C$="[illegible]" THEN LET C$="[illegible]"
1020 IF C$="[illegible]" THEN LET C$="[illegible]"
1030 IF C$="[illegible]" THEN LET C$="[illegible]"
1040 IF C$="[illegible]" THEN LET C$="[illegible]"
1050 IF C$="[illegible]" THEN LET C$="[illegible]"
1060 LET B=INT (RND*4)+1
1070 IF B=1 THEN LET D$="[illegible]"
1080 IF B=2 THEN LET D$="[illegible]"
1090 IF B=3 THEN LET D$="[illegible]"
1100 IF B=4 THEN LET D$="[illegible]"
1110 RETURN
2000 PRINT AT 21,0;"ASSO VALE 1 O 11 (SE 11 BATTI 2)"
2005 IF INKEY$="" THEN GOTO 2005
2010 LET Z=VAL INKEY$
2020 IF Z=2 THEN GOTO 2040
2030 LET D=D-10
2040 LET H=H-1
2050 IF H=1 THEN GOTO 2000
2060 RETURN
3000 IF F+11<=21 OR (F+11=D AND F+11<=21) THEN RETURN
3010 LET C=1
3020 RETURN
```

Derby

Il programma simula una corsa di cavalli. I cavalli sono rappresentati sullo schermo da tre lettere: A, B, C. I cavalli al via compiono due giri dello schermo e alla fine della gara appare la quotazione con cui vengono pagate le scommesse sul vincitore. Questa è una versione abbreviata di un programma apparso in "Getting acquainted with your ZX81", un opuscolo distribuito dalla rivista INTERFACE dedicato ai soli programmi per ZX81. Il lettore che ha proposto la modifica aggiunge alcune osservazioni: "... Lo ZX81 è un'ottima macchina in rapporto al prezzo. La limitazione più evidente riguarda la scarsa quantità di memoria disponibile senza l'espansione RAM. Quando mi sono reso conto di ciò mi sono precipitato ad ordinare un esemplare già ai primi di giugno. In risposta ho ricevuto una lettera in cui la ditta diceva di non poter evadere il mio ordine prima della fine di agosto. In compenso mi veniva garantito che, assieme alla scheda, mi sarebbe stata spedita una cassetta di programmi di dimostrazione dell'utilità della stessa!"

sinclair ZX81

```
   1 REM *** DERBY ***
   5 CLS
   6 PRINT AT 12,10;"D E R B Y"
   7 FOR Z=1 TO 200
   8 NEXT Z
  10 DIM A(3)
  20 CLS
  22 PRINT AT 1,1;"PRONTI..."
  30 FOR N=1 TO 100
  40 NEXT N
  45 PRINT AT 1,1;"...VIA..."
  46 FOR K=1 TO 10
  47 NEXT K
  50 CLS
  60 FOR G=1 TO 3
  70 LET A(G)=1+A(G)+INT (RND*2)
  80 NEXT G
  90 PRINT AT ABS (22-A(1)),12;"
A"
 100 PRINT AT ABS (22-A(2)),15;"
B"
 110 PRINT AT ABS (22-A(3)),18;"
C"
 120 IF A(1)<41 AND A(2)<41 AND
A(3)<41 THEN GOTO 50
 125 PRINT AT 1,9;"TOTALIZZATORE
"
 130 IF A(1)>40 THEN PRINT AT 3,
2;"VINCE A"
 140 IF A(2)>40 THEN PRINT AT 3,
2;"VINCE B"
 150 IF A(3)>40 THEN PRINT AT 3,
2;"VINCE C"
 160 LET X=1+INT (RND*3)
 170 PRINT AT 5,2;"DATO *";X;"*
A 1"
 172 FOR H=1 TO 200
 175 NEXT H
 180 RUN
```

Musica

Questo programmino è un utile trucco per ottenere suoni con lo ZX81. Per sentire la melodia dovrete alzare il volume del televisore impiegato come monitor e spostare la sintonia fino a riuscire ad ascoltare il suono prodotto dalle interferenze sullo schermo prodotte dall'alternarsi rapido delle istruzioni FAST e SLOW.

sinclair ZX81

```
   1 REM *** MUSICA ***
  10 FOR A=1 TO 100
  20 SLOW
  30 FAST
  40 NEXT A
  50 FOR A=1 TO 100
  60 SLOW
  70 FAST
  80 NEXT A
  90 FOR A=1 TO 100
 100 SLOW
 110 FAST
 120 NEXT A
 130 GOTO 10
```

La guida sicura nel labirinto tecnologico.

TechnoClub è l'organizzazione di vendita per corrispondenza del libro tecnico (principalmente elettronica e informatica) nonché del software applicativo.
TechnoClub è anche il tuo consulente, la guida sicura per orientarsi nel labirinto dell'editoria tecnica, lo strumento ed il servizio essenziale per il numero crescente di persone che hanno compreso l'importanza della tecnologia nel mondo odierno.
Libri di base e didattici per imparare a capire; applicativi per realizzare e coltivare il proprio hobby; pratici per risolvere i problemi dell'attività quotidiana; di elevata specializzazione per migliorare il proprio background professionale o culturale. E altri ancora per soddisfare ogni esigenza.
TechnoClub offre solo il meglio della produzione tecnica editoriale. Per questo ha scelto di collaborare con qualificati editori italiani e soprattutto si avvale di un'équipe di professionisti che esamina, seleziona e propone le opere più significative e complete.

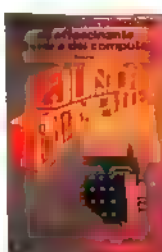
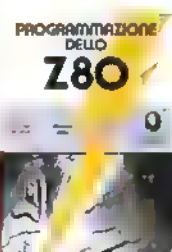
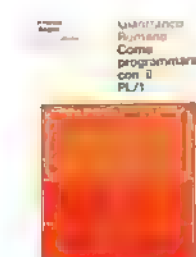
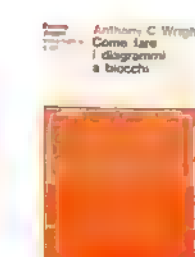
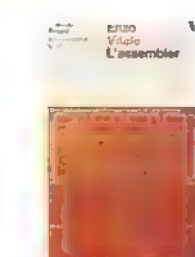
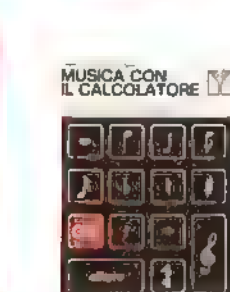
TechnoClub
Via ROSELLINI, 12 MILANO

TechnoClub

TechnoClub ha instaurato rapporti di collaborazione con i più prestigiosi editori e software-house stranieri, per offrire tempestivamente, già da quest'anno, le opere più innovative in lingua originale e il software più interessante, appena disponibili. Tutti possono aderire al TechnoClub, assicurandosi un servizio garantito, professionale, veloce, unico nel suo genere. Esamina le modalità per diventare Socio e considera i numerosi vantaggi che ne derivano.

TechnoClub

i migliori libri tecnici e il software a casa vostra.

Cod. IHC01 — Cod. IHC02 — Cod. IHC03 — Cod. IHC04 — Cod. IHC05 — Cod. IHC06 — Cod. IHC07

Cod. IFC04 — Cod. IFC05 — Cod. IGC01 — Cod. IFH02 — Cod. IFH04 — Cod. IFH07 — Cod. IFH08

Cod. IAK03 — Cod. IAK04 — Cod. IAK05 — Cod. IAK06 — Cod. IAK07 — Cod. IAK08 — Cod. IAK09

Cod. IHK02 — Cod. IHK03 — Cod. IHK04 — Cod. IFK01 — Cod. IFK02 — Cod. IFK03 — Cod. IFK04

Associati subito.
Hai almeno 8 buone ragioni per farlo.

1. **Nessun impegno di acquisto.**
I Soci non sono vincolati all'acquisto di un numero minimo di libri durante il periodo di adesione al **TechnoClub**. Di conseguenza, scelta libera e nessuna imposizione, acquistando quello che si vuole, quando si vuole.

2. **Garanzia.**
I libri proposti dal **TechnoClub** costituiscono sempre la versione originale e più aggiornata delle edizioni in commercio.
Il **TechnoClub** garantisce quindi il contenuto e la veste tipografica originali.

3. **Convenienza certa.**
Il prezzo delle opere offerte ai Soci del **TechnoClub** è inferiore del 10% circa rispetto al prezzo di copertina dell'edizione in commercio. Il risparmio è perciò assicurato.

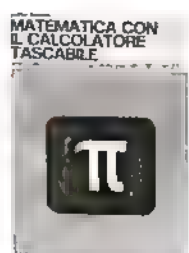
4. **Consulenza professionale per una scelta sicura.**
La selezione delle opere proposte dal **TechnoClub** è effettuata da un gruppo di esperti dei singoli settori.
Viene in tal modo offerto ai Soci un orientamento sicuro e garantita la massima affidabilità nella scelta.

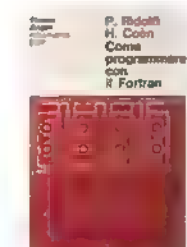
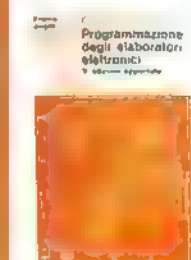
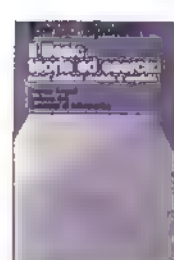
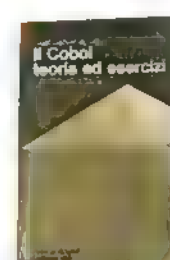
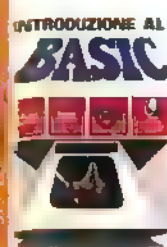
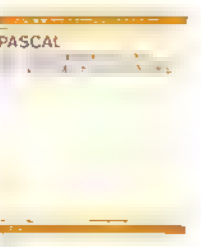
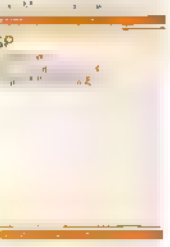
5. **Informazione costante.**
A tutti i soci del **TechnoClub** viene inviata gratuitamente, ogni tre mesi, la rivista **"TechnoClub Review"**, che presenta l'assortimento, suddiviso per argomento e settore specifico di interesse, dei libri selezionati. Ogni libro viene illustrato con note esplicative che ne chiariscono il contenuto.
Il Socio viene in tal modo facilitato nella scelta, secondo le sue specifiche esigenze.

6. **Aggiornamento continuo.**
"TechnoClub Review" garantisce inoltre l'aggiornamento costante sulle novità editoriali.
Considerando l'evoluzione continua dei settori trattati, i Soci dispongono così di uno strumento efficace per tenersi tempestivamente aggiornati.

7. **Un ulteriore e interessante vantaggio.**
I Soci ricevono anche la tessera **TechnoClub**, un documento strettamente personale che dà diritto a sconti speciali sugli acquisti effettuati presso i negozi convenzionati, indicati sulla rivista **"TechnoClub Review"**.

8. **Praticità e comodità d'acquisto.**
Aderire al **TechnoClub** significa poter scegliere con tranquillità a casa propria consultando semplicemente la rivista **"TechnoClub Review"**.
Garanzia di libri sempre disponibili, nessuna perdita di tempo in lunghe ricerche... e i libri arrivano puntualmente a domicilio.

...e puoi già scegliere tra questi titoli.

Cod. IHC08 — Cod. IHC09 — Cod. IFC01 — Cod. IFC02 — Cod. IFC03

Cod. IFH11 — Cod. IFH12 — Cod. IDH02 — Cod. IAK01 — Cod. IAK02

Cod. IAK10 — Cod. IAK14 — Cod. IAK15 — Cod. IAK16 — Cod. IHK01

Cod. IFK05 — Cod. IDK01 — Cod. IDK02 — Cod. IDK03 — Cod. IIK01

32 PROGRAMMI CON IL PET
T. Rugg e P. Feldman - pag. 240, 1981

Trentadue programmi documentati, da eseguire su ogni tipi di PET. Ogni programma si compone di. scopo - come usarlo - esecuzione di prova (con fotografie schermo durante l'esecuzione) - lista del programma, semplici variazioni - routine principali - variabili principali - progetti suggeriti.
Cod. IHC01 **L. 8.500**

32 PROGRAMMI CON L'APPLE
T. Rugg e P. Feldman - pag. 248, 1981

Come sopra, per ogni tipo di Apple
Cod. IHC02 **L. 8.500**

32 PROGRAMMI CON IL TRS-80
T. Rugg e P. Feldman - pag 238, 1981

Come sopra, per il TRS-80.
Cod. IHC03 **L. 8.500**

IMPARATE IL BASIC CON IL PET
H.D. Peckham - pag. 245, 1981

Un corso di autoistruzione per chi desidera imparare il BASIC su un PET. Con esempi ed esercizi, partendo dai concetti più semplici, si arriva a programmare il PET, sfruttando tutte le possibilità.
Cod. IHC04 **L. 8.500**

Come diventare socio...

Per diventare Socio è sufficiente scegliere tra queste due semplici possibilità:

A) Versare l'importo di L. 8.000 quale quota di adesione

B) Effettuare un primo acquisto di libri, per un importo minimo di L 30 000
In questo caso non si versa la quota di adesione
Per acquisti inferiori a L. 30 000 va aggiunta la quota di adesione di L. 8 000

In ambedue i casi, *il Socio ha diritto a ricevere gratuitamente la rivista* **"TechnoClub Review"** *per ben due anni e la tessera personale* con validità per lo stesso periodo
Il Socio che nel corso dei due anni di adesione effettuerà acquisti di libri per un importo di almeno L. 60 000 avrà diritto al rinnovo automatico e gratuito dell'iscrizione al **TechnoClub** per un altro anno, conservando quindi tutti i vantaggi esclusivi

Associati subito.

Spedisci oggi stesso la cedola di adesione

CEDOLA DI ADESIONE da compilare e spedire in busta chiusa a
TechnoClub - Via Rosellini, 12 - 20124 Milano

Si, aderisco al **TechnoClub** scegliendo la seguente formula
☐ A) Sola adesione con versamento di L. 8 000
☐ B) Adesione con acquisto dei seguenti libri per un importo totale di L .... + L. 1 500 per contributo fisso per spese di spedizione

Cod. .... Cod. .... Cod. ....

Cod. .... Cod. .... Cod. ....

☐ Contanti o francobolli allegati
☐ Assegno allegato n° ....
Banca ....
☐ Ho spedito l'importo a mezzo vaglia postale
☐ Ho versato l'importo sul ccp n° 19445204 intestato a **TechnoClub** Milano
☐ Pagherò in contrassegno al postino al ricevimento dei volumi (valido solo per la formula B)

Nome ....
Cognome ....
Via ....
Città .... Cap ....
Cod Fiscale (per le aziende)
Data .... Firma ....

Sono interessato principalmente a Libri di ...
☐ Elettrotecnica
☐ Elettronica e dispositivi elettronici
☐ Elettronica pratica ed hobbystica
☐ Misure elettroniche
☐ Radioriparazioni - TV Service
☐ Equivalenze dei semiconduttori
☐ Personal computer e calcolatrici
☐ Linguaggi e metodi di programmazione
☐ Informatica
☐ Informatica e organizzazione aziendale
☐ Comunicazioni: elementi e sistemi
☐ Microprocessori
☐ Saggistica elettronica e informatica
☐ Energie alternative
☐ Sistemi di regolazione e controllo
☐ Altri (specificare)
....

Sono interessato anche a libri in lingua originale ...
☐ Inglese ☐ Francese ☐ Tedesco
... Sono interessato a Software per .
☐ Apple
☐ Atari
☐ Commodore
☐ Sinclair
☐ Tandy Radio Shack
☐ Altri (specificare)

...e puoi già scegliere tra questi titoli.

INTERVISTA SUL PERSONAL COMPUTER - HARDWARE
R. Didday - pag. 244, 1981

Seicento domande e risposte sul mondo dei personal computer
Questo volume, dedicato all'hardware, contiene un'introduzione, sotto forma di intervista, ai computer in generale e ai microprocessori in particolare.
Cod. IHC05 **L. 8.500**

INTERVISTA SUL PERSONAL COMPUTER - SOFTWARE
R. Didday - pag. 200, 1981

Centinaia di domande e risposte sul mondo dei personal computer. Questo secondo volume, dedicato al software, contiene un'introduzione, sotto forma di intervista, alla programmazione dei computer in generale e dei microprocessori in particolare.
Cod. IHC06 **L. 8.500**

ASTRONOMIA CON IL CALCOLATORE TASCABILE
A. Jones - pag. 307, 1981

Il libro indica come usare i moderni calcolatori tascabili per risolvere diversi problemi astronomici in tempo minimo. Vengono introdotti metodi sia per logica algebrica che per logica polacca inversa.
Appendici con 57 programmi documentati per calcolatori con e senza schedine magnetiche.
Cod. IHC07 **L. 12.000**

LE SCIENZE CON IL CALCOLATORE TASCABILE
D.R. Green e J. Lewis - pag. 398, 1980

Il libro tratta le varie funzioni disponibili sui calcolatori dimostrando la possibilità di applicazione a numerosi problemi di fisica, chimica, biologia, matematica, ingegneria, con diversi esempi svolti e numerosi problemi presi dalle scienze, che il lettore deve svolgere
Cod. IHC08 **L. 9.900**

MATEMATICA CON IL CALCOLATORE TASCABILE
P. Henrici - pag 233, 1980

35 programmi scritti per l'HP-33E e per l'HP-25, che implementano algoritmi di teoria dei numeri, soluzioni di equazioni, teoria della stabilità algebrica, analisi delle serie di potenze, integrazione e funzioni speciali, completati dai diagrammi di flusso e dalle istruzioni operative.
Cod. IHC09 **L. 13.950**

IMPARIAMO A PROGRAMMARE IN BASIC CON IL PET/CBM
R Bonelli - pag 180, 1981

Un corso didattico di programmazione che dalle premesse generali arriva alle frasi BASIC, ai concetti di stringhe, al trattamento dei file, all'uso dei floppy disk, alle norme operative, al DOS, sino al linguaggio macchina.
Cod. IFC01 **L. 9.000**

IMPARIAMO A PROGRAMMARE IN BASIC CON IL VIC/CBM
R. Bonelli - pag. 176, 1981

Un manuale che inizia i principianti alla programmazione in BASIC del VIC20, privilegiando l'aspetto pratico, per imparare nel frattempo che cos'è un programma.
Cod. IFC02 **L. 9.900**

GUIDA AL SINCLAIR ZX81 - ZX80 e NUOVA ROM
R. Bonelli - pag. 264, 1982

Vengono confrontati i tre calcolatori, che presentano notevoli differenze nel sistema di gestione e nel BASIC utilizzato, con considerazioni sulla loro potenzialità e approfondimenti su argomenti specifici: trasformazione dei programmi da un calcolatore all'altro, sistema operativo, gestione dei file, linguaggio macchina.
Cod. IFC03 **L. 14.500**

DAI - MANUALE DEL MICROCOMPUTER
R. Bonelli C. Fiorentini - pag. 145, 1981

La prima parte del manuale insegna l'utilizzo del calcolatore DAI, dai primi passi nella programmazione BASIC alla spiegazione delle caratteristiche di questo personal. La seconda parte contiene le specifiche sull'implementazione del linguaggio BASIC sul calcolatore DAI.
Cod. IFC04 **L. 8.000**

INTRODUZIONE AL PERSONAL E BUSINESS COMPUTING
R.Zaks - pag 224, 1979

Un testo didattico per principianti e per utenti che vogliono ampliare la loro conoscenza.
Un'introduzione ai concetti, alle periferiche e alle tecniche fino ai metodi di valutazione per una scelta oculata.
Cod. IFC05 **L. 12.500**

JUNIOR COMPUTER - Vol. 1
A. Nachtmann e G.H. Nachbar - pag. 178, 1981

Junior Computer è il microelaboratore da autocostruire su un unico circuito stampato. Il sistema base e questo libro sono l'occorrente per l'apprendimento Prossimamente verranno pubblicati altri volumi relativi all'espandibilità del sistema.
Cod. IGC01 **L. 9.900**

PROGRAMMAZIONE DELLO Z-80
R. Zaks - pag. 530, 1981

Libro ideato come testo autonomo e progettato sotto forma di corso per imparare la programmazione in linguaggio Assembler del microprocessore Z-80. dai concetti di base alle tecniche di programmazione più avanzate, con risoluzione obbligatoria di vari esercizi.
Cod. IFH02 **L. 21.500**

PROGRAMMAZIONE DEL 6502
R. Zaks - pag. 375, 1981

Come sopra per il microprocessore 6502
Cod. IFH04 **L. 19.800**

USARE IL MICROPROCESSORE
G. Giaccaglini - pag. 295, 1981

Un testo per far capire l'utilizzo più razionale del microprocessore, soprattutto in relazione al controllo di impianti e processi, con diversi esempi di simulazione e prospettando problemi di interfacciamento hardware.
Cod. IFH07 **L. 13.500**

APPLICAZIONI DEL 6502
R. Zaks - pag. 214, 1981

Tecniche e programmi per applicazioni tipiche del 6502. I programmi sono, con poche varianti, applicabili direttamente su qualunque microcomputer su scheda basato sul 6502, quali il KYM, il SYM e l'AIM 65 e altri e consentono al lettore alcune realizzazioni pratiche.
Cod. IFH08 **L. 12.000**

INTRODUZIONE AI MICROCOMPUTER VOL. 0 IL LIBRO DEL PRINCIPIANTE
A. Osborne - pag. 240, 1980

Una visione complessiva su calcolatori ed elaboratori, con concetti generali e terminologia di base per capire la tecnologia usata. Vengono illustrate le singole parti del sistema, con le possibilità di espansione e componenti accessori.
Cod. IFH11 **L. 12.500**

INTRODUZIONE AI MICROCOMPUTER VOL. 1 IL LIBRO DEI CONCETTI FONDAMENTALI
A. Osborne - pag. 321, 1980

Il libro presenta la struttura logica fondamentale su cui sono basati i sistemi a microcomputer. Usando i concetti comuni a ogni sistema a microprocessore, viene illustrata l'architettura, la programmazione, le possibilità e l'operatività di un microcomputer, con un set finale ipotetico di istruzioni per la simulazione delle possibili situazioni reali in cui si verrà a trovare con i vari microprocessori.
Cod. IFH12 **L. 14.400**

MICROPROCESSORI 8080 e BUS MMS-8
a cura di L'EMMECI - pag. 160, 1981

Un testo di "consulenza" con caratteristiche didattiche per chi ha già una conoscenza di base su circuiti logici, algebra di Boole e aritmetica binaria. Inteso come manuale commentato del microprocessore 8080 e del bus adottato presso il laboratorio di microcalcolatori del Politecnico di Milano Si pone come riferimento per chi programma nel linguaggio Assembler 8080
Cod. IDH02 **L. 5.400**

INTRODUZIONE ALL'APL E ALLE SUE APPLICAZIONI
C. Camerata - pag. 266, 1980

Il linguaggio di programmazione APL si basa su funzioni logico-matematiche di tipo generale. Adatto quindi ad applicazioni in vari settori, è stato implementato su più sistemi per tutti i livelli di elaboratori. Il libro si prefigge lo scopo di fornire agli utenti una conoscenza generale di questo linguaggio
Cod. IAK01 **L. 10.800**

COME PROGRAMMARE CON IL FORTRAN
P Ridolfi/H. Coen - pag. 148, 1980

Una guida allo studio del Fortran, concepita con intenti prevalentemente didattici, che pone in grado di scrivere e capire semplici programmi utilizzando questo "linguaggio simbolico", particolarmente adatto per agevolare gli scienziati e i tecnici all'impiego dell'elaboratore elettronico.
Cod. IAK02 **L. 5.400**

IL FORTRAN - TEORIA ED ESERCIZI
P. Ridolfi - pag. 168, 1981

Viene descritto il complesso di regole che costituiscono la grammatica del Fortran, con abbondanti esemplificazioni pratiche, onde acquisire una graduale e completa padronanza del linguaggio. Non si richiede al lettore una conoscenza approfondita degli elaboratori elettronici
Cod. IAK03 **L. 4.950**

ESERCITAZIONI DI ASSEMBLER
E. Vitale/A. Hernandez - pag. 134, 1981

Una raccolta di 12 esercizi di crescente difficoltà, di ognuno dei quali viene dato l'enunciato, il diagramma a blocchi e la soluzione; quest'ultima costituita dalle liste di compilazioni integrali per ogni problema enunciato.
Cod. IAK04 **L. 4.950**

COME PROGRAMMARE CON IL PL/1
G. Romano - pag. 205, 1982

Il volume presenta le caratteristiche più importanti del linguaggio PL/1, che può definirsi un ibrido tra i linguaggi per utenti scientifici e quelli per utenti commerciali.
Con il ricorso ad esempi esplicativi, si pone come guida per lo studio autodidattico o anche guidato del PL/1.
Cod. IAK05 **L. 7.200**

COME FARE I DIAGRAMMI A BLOCCHI
A.C. Wright - pag. 84, 1981

Con un'abbondante documentazione di esemplificazioni, il libro si propone l'agevole assimilazione della tecnica della diagrammazione a blocchi attraverso numerosi esercizi da svolgere per proprio conto.
Cod. IAK06 **L. 4.500**

L'ASSEMBLER
E. Vitale - pag. 172, 1981

L'opera, rivolta a chi abbia già qualche conoscenza di programmazione, illustra il flessibile linguaggio Assembler, esponendo le varie istruzioni e spiegandone il significato, le modalità d'uso e i casi in cui possono rivelarsi utili o necessarie.
Cod. IAK07 **L. 6.750**

COME PROGRAMMARE CON l'RPG I - RPG II - RPG III
A.C. Wright - pag. 266, 1981

Nella prima parte del testo viene descritto l'RPG I: la sua logica, l'uso di unità periferiche facili da gestire, le maschere di stampa, ecc. Nella seconda e terza parte vengono descritte le istruzioni aggiuntesi nelle più recenti versioni di questo linguaggio: l'RPG II e il compilatore RPG III.
Cod. IAK08 **L. 10.800**

COME PROGRAMMARE CON IL COBOL
G. Sarri - pag. 129, 1982

L'opera espone le regole da seguire per redigere un programma in Cobol, con numerose esemplificazioni e si propone di porre il lettore in grado di scrivere programmi in Cobol e di capirli.
Cod. IAK09 **L. 4.500**

PROGRAMMAZIONE DEGLI ELABORATORI ELETTRONICI
G. Bertoldi - pag. 286, 1982

La prima parte illustra i principi generali dell'elaborazione automatica dei dati, le caratteristiche e il funzionamento degli elaboratori e i problemi di programmazione. La seconda parte è dedicata all'analisi (studio e definizione dei problemi da svolgere); la terza parte illustra la tecnica di stesura dei diagrammi a blocchi. Segue capitolo sul Cobol e uno sui sistemi operativi, più glossario in appendice.
Cod. IAK10 **L. 7.200**

PRINCIPI DI PROGETTAZIONE DEI COMPILATORI
D. Gries - pag. 566, 1980

Il volume descrive le tecniche utilizzabili nella scrittura di compilatori per linguaggi ad alto livello, quali il Fortran ed il PL/1, illustrando teoria ed aspetti pratici connessi con tale scrittura.
Cod. IAK14 **L. 26.100**

IL BASIC - TEORIA ED ESERCIZI
E. Spoletini - pag. 298, 5ª ediz. 1982

Il volume, rivolto anche ai "non addetti ai lavori", presenta le caratteristiche del linguaggio BASIC. L'impostazione didattica si prefigge di far apprendere agevolmente il linguaggio per gradi. Quasi tutti gli esercizi, differenziati come argomenti e grado di difficoltà, sono corredati da diagrammi a blocchi.
Cod. IAK15 **L. 10.800**

IL COBOL - TEORIA ED ESERCIZI
E. Spoletini - pag. 396, 1982

La prima parte dell'opera espone gli elementi per scrivere un programma di media difficoltà; nella seconda parte vengono trattati problemi relativi alle tabelle e ai flussi organizzati in modo non sequenziale. Ogni capitolo è corredato di numerosi esempi ed esercizi.
Cod. IAK16 **L. 13.500**

IL PROGETTO DEI MICROCOMPUTER: SOFTWARE
C.A. Ogdin - pag. 247, 1981

Vengono esposte tecniche aggiornate, tra cui la programmazione strutturata, con diversi esempi di progettazione software che utilizza una notazione simile al Pascal. Viene dato risalto sia al progetto di strutture dati che di procedure strutturate.
Cod. IHK01 **L. 12.150**

PASCAL
P.M. Chirlian - pag. 200, 1981

Questo libro, inteso come manuale di autoistruzione o libro di testo in un corso, per chi non ha esperienza di calcolatori o programmazione, presenta il linguaggio Pascal che permette la "programmazione strutturata". Ogni capitolo si conclude con una serie di esercizi.
Cod. IHK02 **L. 7.650**

MICROSOFT BASIC
K. Knecht - pag. 150, 1981

Un manuale di introduzione al Microsoft BASIC, sorto dall'esigenza di standardizzazione del BASIC per l'implementazione su una varietà di personal computer. Viene dato rilievo alle diverse caratteristiche e forme che il Microsoft Basic può presentare e viene dato particolare risalto alla versione implementata sul TRS-80.
Cod. IHK03 **L. 5.850**

MUSICA CON IL CALCOLATORE
R.C. Zaripov - pag. 169, 1979

Una monografia dedicata al problema della composizione di musica con l'aiuto di calcoli matematico-probabilistici, con rassegna degli studi svolti nel mondo sull'aiuto dei computer per la composizione e l'analisi della musica, oltre alle regole trovate dall'autore per realizzare un modello che simula l'attività di un compositore.
Cod. IHK04 **L. 6.750**

CP/M CON MP/M
R. Zaks - pag. 309, 1982

Il libro si prefigge di rendere agevole l'uso del CP/M (nelle versioni CP/M 1.4 - CP/M 2.2 - sistema operativo multiutente MP/M); il sistema operativo progettato per semplificare l'utilizzo di un microcomputer, disponibile su quasi tutti gli elaboratori basati su microprocessore 8080 e Z80 e su certi sistemi utilizzanti il 6502.
Cod. IFK01 **L. 19.800**

PROGRAMMARE IN ASSEMBLER
A. Pinaud - pag. 153, 1982

Il libro, destinato in particolare a chi già ha una buona conoscenza di un linguaggio evoluto molto semplice come il BASIC, fornisce i rudimenti che consentono di programmare in Assembler, con numerosi esempi pratici. Come Assembler esistente è stato scelto quello dello Z80.
Cod. IFK02 **L. 9.000**

IMPARIAMO IL PASCAL
F. Waldner - pag. 162, 1981

Un libro di divulgazione, incentrato sull'autoapprendimento del linguaggio Pascal, con consigli, problemi.
Un testo da "usare" e non da "leggere", secondo l'intento dichiarato dall'autore.
Cod. IFK03 **L. 9.000**

PASCAL-MANUALE E STANDARD DEL LINGUAGGIO
K. Jensen/N. Wirth - pag. 179, 1981

La prima parte è costituita dal manuale di apprendimento delle regole e dei principi del Pascal (manuale utente); la seconda, dalla descrizione dello standard del linguaggio. Serie di tavole contenenti la definizione del Pascal attraverso il BNF e anche tramite le carte sintattiche.
Cod. IFK04 **L. 9.000**

INTRODUZIONE AL BASIC
P. Le Beux - pag. 314, 1981

Un corso rivolto ai principianti, che illustra tutti gli aspetti del BASIC su differenti sistemi. Con numerosi esempi, il lettore può verificare con immediatezza il reale apprendimento raggiunto.
Cod. IFK05 **L. 16.500**

METODI DI OTTIMIZZAZIONE E PROGRAMMI DI CALCOLO
C. Baldissera/S. Ceri/A. Colorni - pag. 468, 1981

Vengono esposte le caratteristiche di un gruppo di programmi (attualmente esistente presso il Centro di Teoria dei sistemi del CNR), che si compone di alcuni classici algoritmi strutturati per privilegiare la semplicità d'uso, l'uniformità e modularità fra i vari programmi. Il linguaggio usato è il Fortran.
Cod. IDK01 **L. 18.000**

PASCAL - DAL MICROPROCESSORE AL GRANDE ELABORATORE
G. Cioni/S. Crespi Reghizzi/M. Moscarini
pag. 194, 1981

Il testo, rivolto a chi già conosce la programmazione del calcolatore, descrive nella prima parte il linguaggio Pascal standard nelle sue varie parti, con 40 esempi di programmi, collaudati sul calcolatore.
Nella seconda parte, dopo un cenno alle tecniche di compilazione, si presentano le schede tecniche di alcune versioni del Pascal disponibili per elaboratori di vasta diffusione.
Cod. IDK02 **L. 7.200**

LISP - LINGUAGGIO E METODOLOGIA DI PROGRAMMAZIONE
G. Gini, M. Gini, G. Guida - pag. 204, 1981

Con molti esempi di diversa complessità nel testo vengono illustrati tutti gli aspetti del linguaggio LISP, oggi diffuso anche su mini e personal computer, utilizzato per progettare sistemi di intelligenza artificiale e per la soluzione di problemi di elaborazione non numerica.
Cod. IDK03 **L. 8.100**

PROGRAMMARE IN FORTRAN 375 PROBLEMI RISOLTI
S. Lipschutz/A. Poe - pag. 314, 1980

Lo scopo del libro è di presentare il linguaggio Fortran e di insegnare a risolvere dei problemi con esso. Oltre alla sintassi, viene insegnato come scrivere dei programmi Fortran con chiarezza, insistendo sia sulle tecniche che sulle buone abitudini di programmazione. Vengono esaminati i principi più importanti sia del Fortran standard che del Fortran strutturato.
Cod. IIK01 **L. 10.800**

**UNA PUBBLICAZIONE
DEL GRUPPO EDITORIALE JACKSON**

PERSONAL SOFTWARE

ANNO 1 N. 2 SETTEMBRE 1982

DIRETTORE RESPONSABILE: Giampietro Zanga

DIRETTORE: Mauro Boscarol

REDAZIONE: Completo Software

HANNO COLLABORATO A QUESTO NUMERO:
P. Dell'Orco, A.Giovannetti, G. Williams, R. Baker, W. Douglas Maurer, J.S. Browning, C. Sintini, P. Canevarolo, N. Tonolli, G. Staluppi, Lexikon
Copertina: Walter Paggioro
Grafica e Impaginazione: Fernanda Arzenton
Fotocomposizione: Composizioni Grafiche - Padova
Traduzioni: F. Santini, R. Pevarolo

CONTABILITÀ: Franco Mancini, Roberto Ostelli, Mariella Luciano, Franca Anelli, Sandra Cicuta, Gabriella Napoli

DIFFUSIONE E ABBONAMENTI: Luigi De Cao, Adela Bel Lozano, Ombretta Giannetto

AUTORIZZAZIONE ALLA PUBBLICAZIONE: Tribunale di Milano n. 69 del 20/2/1982

PUBBLICITÀ: Concessionario per l'Italia e l'Estero
Reina s.r.l. Via Washington, 50 - 20146 Milano
Tel. (02) 4988066/7/8/9/060 (5 linee r.a.)
Telex 316213 REINA I

STAMPA: Arti Grafiche "La Cittadella" S.p.a.
Pieve del Cairo (PV)

Concessionario esclusivo per la DIFFUSIONE in Italia e all'Estero:
SODIP - Via Zuretti, 25 - 20125 Milano

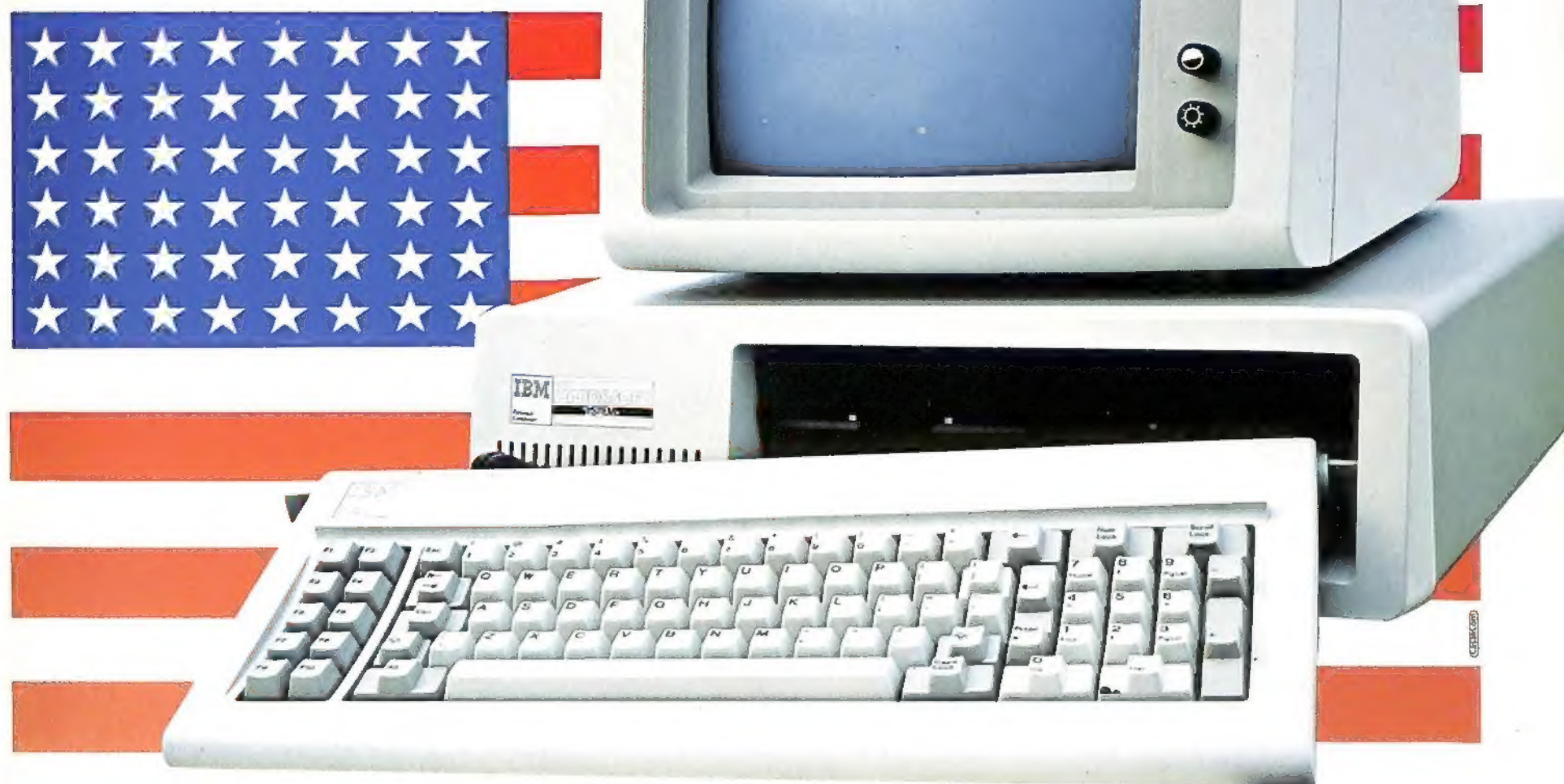
Spedizione in abbonamento Postale Gruppo III/70
Prezzo della rivista L. 3.500. Numero arretrato L. 6.000.
Abbonamento annuo (10 numeri) L. 30.000; per l'Estero L. 48.000
I versamenti vanno indirizzati a: Gruppo Editoriale Jackson - Via Rosellini, 12 - 20124 Milano - mediante emissione di assegno bancario, cartolina vaglia o utilizzando il c/c Postale numero 11666203.
Per i cambi di indirizzo, indicare, oltre naturalmente al nuovo, anche l'indirizzo precedente, ed allegare alla comunicazione l'importo di L. 500, anche in francobolli.

© TUTTI I DIRITTI DI RIPRODUZIONE O TRADUZIONE DEGLI ARTICOLI PUBBLICATI SONO RISERVATI

GRUPPO EDITORIALE JACKSON Srl

DIREZIONE, REDAZIONE, AMMINISTRAZIONE:
Via Rosellini, 12 - 20124 Milano - Telefoni: 68.03.68 - 68.00.54

SEDE LEGALE: Via Vincenzo Monti, 15 - 20123 Milano

DIREZIONE EDITORIALE: Giampietro Zanga e Paolo Reina
COORDINAMENTO EDITORIALE: Daniele Comboni

PERSONAL IBM

IBM PERSONAL COMPUTER: Unità di sistema fino a 256 Kbyte, con microprocessore 8088 a 16 bit. 2 unità drive per minifloppy da 160 Kbyte ciascuno. Monitor fosfori verdi 12", 25 righe per 80 col. Stampante. Alimentatore.

Importato e distribuito da

HARD&SOFT

SYSTEMS

CENTRI DI DISTRIBUZIONE: HARD & SOFT SYSTEM S.R.L., Via Costantinopoli 50, 47045 Miramare di Rimini, Tel. 0541-31060/759076/759077 ● AVELCO S.R.L., Via Cornigliano 47, Genova, Tel. 010-602994 ● DP INOFSHOP S.N.C., V.le Storchi 10/10A, 41100 Modena, Tel. 059-218821 ● SIRIO SHOP, V.le Certosa 148, Milano, Tel. 02-304713/305778/3080786 ● STUDIO P, Via Paganino Bonafede 2/A, 40139 Bologna, Tel. 051-548080 ● INFODIS S.R.L., Via A. Oriani 10, 04100 Latina, Tel. 0773-491271 ● INFORMATICA MERIDIONALE, Via Pigna 92, Napoli, Tel. 081/248988

Apple cresce.

response

Apple ha introdotto il concetto di personal in tutto il mondo. E in tutto il mondo Apple cresce. Cresce anche in Italia dove la Iret, che lo importa e ne cura l'assistenza, può oggi annunciare l'esistenza di una rete di vendita di oltre 200 centri specializzati che fanno di Apple il loro cavallo di battaglia.

Ma cresce anche la gamma Apple. Oltre al già famoso e collaudatissimo Apple II, la Iret presenta Apple III, piú potente e adatto ad usi specialistici. E poi video per ogni esigenza, a fosfori verdi o a colori, stampanti e decine di accessori e programmi.

E naturalmente crescono le vendite di Apple, perché il personal computing conquista piccole aziende, professionisti e privati. È facile prevedere quindi che Apple continuerà a crescere.

apple® computer

Distribuzione per l'Italia

IRET® *informatica*

Via Bovio, 5 - 42100 Reggio Emilia - Tel. 0522/32643 - TLX 530173 IRETRE